

Derin Öğrenme Deep Learning

Hazırlayan: M. Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliği Bölümü

Bu dersin sunumları, "Simon Haykin, Neural Networks and Learning Machines, Prentice Hall, 2016." kitabı kullanılarak hazırlanmıştır.

İçerik

- ▶ Training
- ▶ Değerlendirme
- ▶ CNN mimarileri
- ▶ CNN uygulamaları

Training

- ▶ **CNN'lerin eğitimi**, ANN eğitimi gibi **gradient descent ile backpropagation yöntemi kullanılarak yapılır.**
- ▶ Bir **CNN aşağıdaki gibi tanımlanan bir fonksiyondur.**

$$o = f(x, w)$$

- ▶ Burada, x görüntüleri, w ağ parametrelerini (ağırlıklar, bias), o softmax çıkışı gösterir.
- ▶ Eğitim sürecinde çıkıştaki hatayı minimize edecek ağırlıklar hesaplanır.

$$E(y, f(x, w))$$

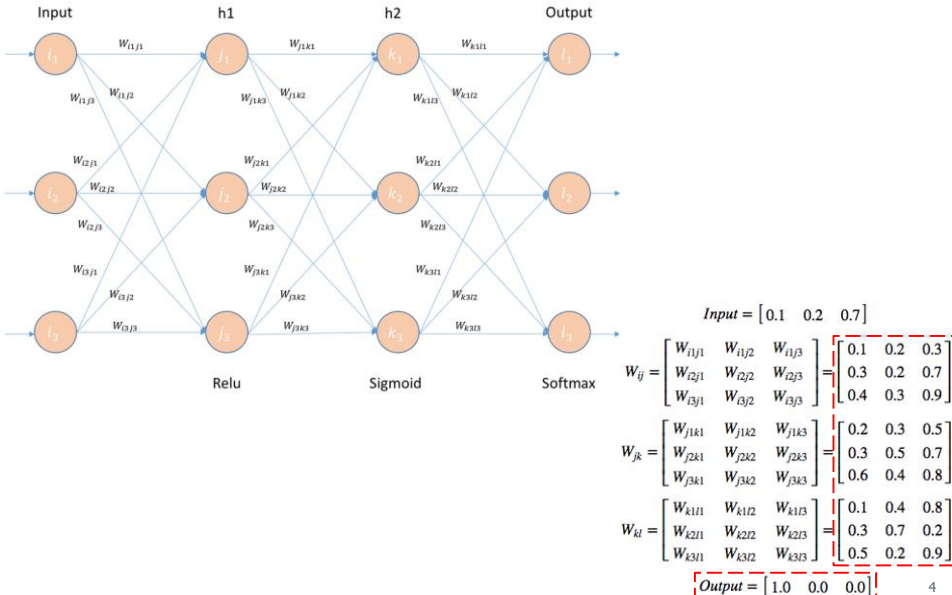
- ▶ Burada, y beklenen çıkışı gösterir.
- ▶ Ağırlıklar gradient descent ile iteratif bir şekilde değiştirilir.

$$w(k+1) = w(k) + \eta \frac{\partial E}{\partial w}$$

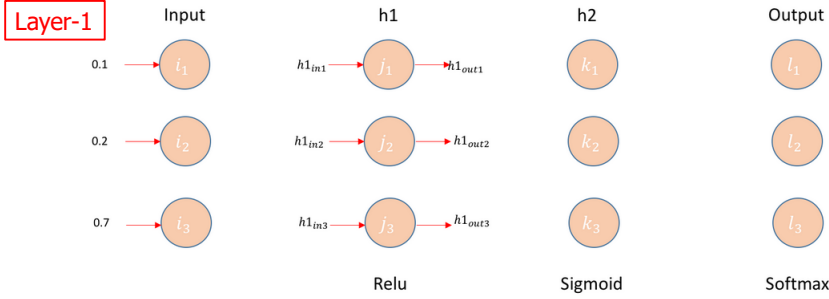
3

Training – İleri geçiş

- ▶ Aşağıdaki ağ için ağırlıklar ve bias değerleri rastgele başlatılır.



Training – İleri geçiř



Matris iřlemi

$$\begin{bmatrix} i_1 & i_2 & i_3 \end{bmatrix} \times \begin{bmatrix} W_{i1j1} & W_{i1j2} & W_{i1j3} \\ W_{i2j1} & W_{i2j2} & W_{i2j3} \\ W_{i3j1} & W_{i3j2} & W_{i3j3} \end{bmatrix} + \begin{bmatrix} b_{j1} & b_{j2} & b_{j3} \end{bmatrix} = \begin{bmatrix} h_{1in1} & h_{1in2} & h_{1in3} \end{bmatrix}$$

$$relu = \max(0, x)$$

ReLU iřlemi

$$\begin{bmatrix} h_{1out1} & h_{1out2} & h_{1out3} \end{bmatrix} = \begin{bmatrix} \max(0, h_{1in1}) & \max(0, h_{1in2}) & \max(0, h_{1in3}) \end{bmatrix}$$

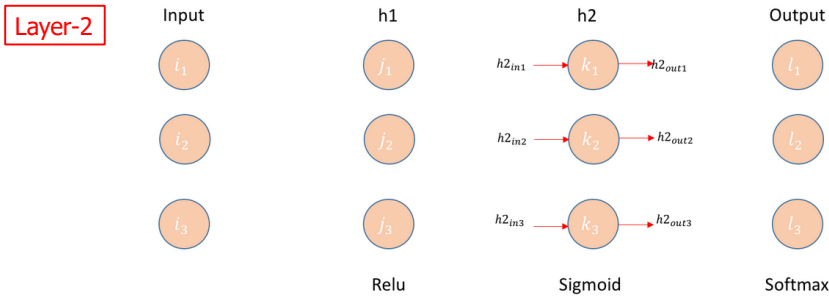
Örnek

$$\begin{bmatrix} 0.1 & 0.2 & 0.7 \end{bmatrix} \times \begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0.3 & 0.2 & 0.7 \\ 0.4 & 0.3 & 0.9 \end{bmatrix} + \begin{bmatrix} 1.0 & 1.0 & 1.0 \end{bmatrix} = \begin{bmatrix} 1.35 & 1.27 & 1.8 \end{bmatrix}$$

$$\begin{bmatrix} h_{1out1} & h_{1out2} & h_{1out3} \end{bmatrix} = \begin{bmatrix} 1.35 & 1.27 & 1.8 \end{bmatrix}$$

5

Training – İleri geçiř



Matris iřlemi

$$\begin{bmatrix} h_{1out1} & h_{1out2} & h_{1out3} \end{bmatrix} \times \begin{bmatrix} W_{j1k1} & W_{j1k2} & W_{j1k3} \\ W_{j2k1} & W_{j2k2} & W_{j2k3} \\ W_{j3k1} & W_{j3k2} & W_{j3k3} \end{bmatrix} + \begin{bmatrix} b_{k1} & b_{k2} & b_{k3} \end{bmatrix} = \begin{bmatrix} h_{2in1} & h_{2in2} & h_{2in3} \end{bmatrix}$$

$$Sigmoid = 1/(1 + e^{-x})$$

Sigmoid iřlemi

$$\begin{bmatrix} h_{2out1} & h_{2out2} & h_{2out3} \end{bmatrix} = \begin{bmatrix} 1/(1 + e^{-h_{2in1}}) & 1/(1 + e^{-h_{2in2}}) & 1/(1 + e^{-h_{2in3}}) \end{bmatrix}$$

Örnek

$$\begin{bmatrix} 1.35 & 1.27 & 1.8 \end{bmatrix} \times \begin{bmatrix} 0.2 & 0.3 & 0.5 \\ 0.3 & 0.5 & 0.7 \\ 0.6 & 0.4 & 0.8 \end{bmatrix} + \begin{bmatrix} 1.0 & 1.0 & 1.0 \end{bmatrix} = \begin{bmatrix} 2.73 & 2.76 & 4.001 \end{bmatrix}$$

$$\begin{bmatrix} h_{2out1} & h_{2out2} & h_{2out3} \end{bmatrix} = \begin{bmatrix} 0.938 & 0.94 & 0.98 \end{bmatrix}$$

6

Training – İleri geçiř

Layer-3

Input	h1	h2	Output
i_1	j_1	k_1	$o_{in1} \rightarrow l_1 \rightarrow o_{out1}$
i_2	j_2	k_2	$o_{in2} \rightarrow l_2 \rightarrow o_{out2}$
i_3	j_3	k_3	$o_{in3} \rightarrow l_3 \rightarrow o_{out3}$

Relu Sigmoid Softmax

Matris iřlemi $[h_{2out1} \ h_{2out2} \ h_{2out3}] \times \begin{bmatrix} W_{k1l1} & W_{k1l2} & W_{k1l3} \\ W_{k2l1} & W_{k2l2} & W_{k2l3} \\ W_{k3l1} & W_{k3l2} & W_{k3l3} \end{bmatrix} + [b_{l1} \ b_{l2} \ b_{l3}] = [O_{in1} \ O_{in2} \ O_{in3}]$

$Softmax_{O_{out1}} = e^{O_{in1}} / (\sum_{a=1}^3 e^{O_{ina}})$

Softmax iřlemi $[O_{out1} \ O_{out2} \ O_{out3}] = [e^{O_{in1}} / (\sum_{a=1}^3 e^{O_{ina}}) \ e^{O_{in2}} / (\sum_{a=1}^3 e^{O_{ina}}) \ e^{O_{in3}} / (\sum_{a=1}^3 e^{O_{ina}})]$

Örnek $[0.938 \ 0.94 \ 0.98] \times \begin{bmatrix} 0.1 & 0.4 & 0.8 \\ 0.3 & 0.7 & 0.2 \\ 0.5 & 0.2 & 0.9 \end{bmatrix} + [1.0 \ 1.0 \ 1.0] = [1.8658 \ 2.2292 \ 2.8204]$

$[O_{out1} \ O_{out2} \ O_{out3}] = [0.2698 \ 0.3223 \ 0.4078]$

Training – Backpropagation

Hata

İstenen çıkıř (y) = [1.0 0.0 0.0]
 Hesaplanan çıkıř (O_{out}) = [0.2698 0.3223 0.4078]

$crossentropy = -(\sum_{i=1}^3 (y_i \times \log(O_{outi})) + ((1 - y_i) \times \log((1 - O_{outi}))))$

Örnek

$Error = -((1 * \log(0.2698) + 0 + 0 * \log(0.3223)) + 1 * \log(1 - 0.3223) + 0 * \log(0.4078) + 1 * \log(1 - 0.4078))$
 $= -\log(0.2698) - \log(0.6777) - \log(0.5922)$
 $= +0.569858 + 0.16886 + 0.22753$
 $= 0.97$

Training – Backpropagation

Backpropagation için türevler

Sigmoid

$$\text{Sigmoid} = 1/(1 + e^{-x})$$

$$\frac{\partial \text{Sigmoid}}{\partial x} = \text{Sigmoid} \times (1 - \text{Sigmoid})$$

ReLU

$$\text{relu} = \max(0, x)$$

$$\text{if } x > 0, \frac{\partial(\text{relu})}{\partial x} = 1$$

$$\text{Otherwise, } \frac{\partial(\text{relu})}{\partial x} = 0$$

Softmax

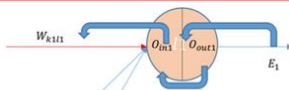
$$\text{Softmax} = e^{x_a} / \left(\sum_{a=1}^n e^{x_a} \right) = e^{x_1} / (e^{x_1} + e^{x_2} + e^{x_3})$$

$$\frac{\partial(\text{Softmax})}{\partial x_1} = (e^{x_1} \times (e^{x_2} + e^{x_3})) / (e^{x_1} + e^{x_2} + e^{x_3})^2$$

9

Training – Backpropagation

Hatanın geri yayılımı (Gizli katman 2 ve çıkış katmanı arasındaki ağırlıklar için)



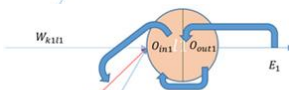
$$\frac{\partial E_1}{\partial O_{out1}} = \frac{\partial(-1 * ((y_1 * \log(O_{out1}) + (1 - y_1) * \log((1 - O_{out1}))))}{\partial O_{out1}}$$

$$= -1 * ((y_1 * (1/O_{out1}) + (1 - y_1) * (1/(1 - O_{out1})))$$



$$\frac{\partial O_{out1}}{\partial O_{in1}} = \frac{\partial(e^{O_{in1}} / (e^{O_{in1}} + e^{O_{in2}} + e^{O_{in3}}))}{\partial O_{in1}}$$

$$= (e^{O_{in1}} \times (e^{O_{in2}} + e^{O_{in3}})) / (e^{O_{in1}} + e^{O_{in2}} + e^{O_{in3}})^2$$



$$\frac{\partial O_{in1}}{\partial W_{k111}} = \frac{\partial((h_{2-out1} * W_{k111}) + (h_{2-out2} * W_{k211}) + (h_{2-out3} * W_{k311}) + b_{11})}{\partial W_{k111}}$$

$$= h_{2-out1}$$



10

Training – Backpropagation

Hatanın geri yayılımı (Gizli katman 2 ve çıkış katmanı arasındaki ağırlıklar için)

$$\begin{bmatrix} \frac{\partial E_1}{\partial O_{out1}} \\ \frac{\partial E_2}{\partial O_{out2}} \\ \frac{\partial E_3}{\partial O_{out3}} \end{bmatrix} = \begin{bmatrix} -1 * ((y_1 * (1/O_{out1}) + (1 - y_1) * (1/(1 - O_{out1}))) \\ -1 * ((y_2 * (1/O_{out2}) + (1 - y_2) * (1/(1 - O_{out2}))) \\ -1 * ((y_3 * (1/O_{out3}) + (1 - y_3) * (1/(1 - O_{out3}))) \end{bmatrix} = \begin{bmatrix} -3.70644 \\ -1.4755 \\ -1.6886 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial O_{out1}}{\partial O_{in1}} \\ \frac{\partial O_{out2}}{\partial O_{in2}} \\ \frac{\partial O_{out3}}{\partial O_{in3}} \end{bmatrix} = \begin{bmatrix} (e^{O_{in1}} \times (e^{O_{in2}} + e^{O_{in3}})) / (e^{O_{in1}} + e^{O_{in2}} + e^{O_{in3}})^2 \\ (e^{O_{in2}} \times (e^{O_{in1}} + e^{O_{in3}})) / (e^{O_{in1}} + e^{O_{in2}} + e^{O_{in3}})^2 \\ (e^{O_{in3}} \times (e^{O_{in1}} + e^{O_{in2}})) / (e^{O_{in1}} + e^{O_{in2}} + e^{O_{in3}})^2 \end{bmatrix} = \begin{bmatrix} 0.15911 \\ 0.2040 \\ 0.3685 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial O_{in1}}{\partial W_{k111}} \\ \frac{\partial O_{in1}}{\partial W_{k211}} \\ \frac{\partial O_{in1}}{\partial W_{k311}} \end{bmatrix} = \begin{bmatrix} h2_{out1} \\ h2_{out2} \\ h2_{out3} \end{bmatrix} = \begin{bmatrix} 0.938 \\ 0.94 \\ 0.98 \end{bmatrix} \quad \begin{bmatrix} \frac{\partial O_{in2}}{\partial W_{k112}} \\ \frac{\partial O_{in2}}{\partial W_{k212}} \\ \frac{\partial O_{in2}}{\partial W_{k312}} \end{bmatrix} = \begin{bmatrix} h2_{out1} \\ h2_{out2} \\ h2_{out3} \end{bmatrix} = \begin{bmatrix} 0.938 \\ 0.94 \\ 0.98 \end{bmatrix} \quad \begin{bmatrix} \frac{\partial O_{in3}}{\partial W_{k113}} \\ \frac{\partial O_{in3}}{\partial W_{k213}} \\ \frac{\partial O_{in3}}{\partial W_{k313}} \end{bmatrix} = \begin{bmatrix} h2_{out1} \\ h2_{out2} \\ h2_{out3} \end{bmatrix} = \begin{bmatrix} 0.938 \\ 0.94 \\ 0.98 \end{bmatrix}$$

11

Training – Backpropagation

Hatanın geri yayılımı (Gizli katman 2 ve çıkış katmanı arasındaki ağırlıklar için)

$$\frac{\partial E_1}{\partial W_{k111}} = \frac{\partial E_1}{\partial O_{out1}} * \frac{\partial O_{out1}}{\partial O_{in1}} * \frac{\partial O_{in1}}{\partial W_{k111}}$$

$$\delta W_{kl} = \begin{bmatrix} \frac{\partial E_1}{\partial W_{k111}} & \frac{\partial E_2}{\partial W_{k112}} & \frac{\partial E_3}{\partial W_{k113}} \\ \frac{\partial E_1}{\partial W_{k211}} & \frac{\partial E_2}{\partial W_{k212}} & \frac{\partial E_3}{\partial W_{k213}} \\ \frac{\partial E_1}{\partial W_{k311}} & \frac{\partial E_2}{\partial W_{k312}} & \frac{\partial E_3}{\partial W_{k313}} \end{bmatrix} = \begin{bmatrix} \frac{\partial E_1}{\partial O_{out1}} * \frac{\partial O_{out1}}{\partial O_{in1}} * \frac{\partial O_{in1}}{\partial W_{k111}} & \frac{\partial E_2}{\partial O_{out2}} * \frac{\partial O_{out2}}{\partial O_{in2}} * \frac{\partial O_{in2}}{\partial W_{k112}} & \frac{\partial E_3}{\partial O_{out3}} * \frac{\partial O_{out3}}{\partial O_{in3}} * \frac{\partial O_{in3}}{\partial W_{k113}} \\ \frac{\partial E_1}{\partial O_{out1}} * \frac{\partial O_{out1}}{\partial O_{in1}} * \frac{\partial O_{in1}}{\partial W_{k211}} & \frac{\partial E_2}{\partial O_{out2}} * \frac{\partial O_{out2}}{\partial O_{in2}} * \frac{\partial O_{in2}}{\partial W_{k212}} & \frac{\partial E_3}{\partial O_{out3}} * \frac{\partial O_{out3}}{\partial O_{in3}} * \frac{\partial O_{in3}}{\partial W_{k213}} \\ \frac{\partial E_1}{\partial O_{out1}} * \frac{\partial O_{out1}}{\partial O_{in1}} * \frac{\partial O_{in1}}{\partial W_{k311}} & \frac{\partial E_2}{\partial O_{out2}} * \frac{\partial O_{out2}}{\partial O_{in2}} * \frac{\partial O_{in2}}{\partial W_{k312}} & \frac{\partial E_3}{\partial O_{out3}} * \frac{\partial O_{out3}}{\partial O_{in3}} * \frac{\partial O_{in3}}{\partial W_{k313}} \end{bmatrix}$$

$$\delta W_{kl} = \begin{bmatrix} \delta W_{k111} & \delta W_{k112} & \delta W_{k113} \\ \delta W_{k211} & \delta W_{k212} & \delta W_{k213} \\ \delta W_{k311} & \delta W_{k312} & \delta W_{k313} \end{bmatrix} = \begin{bmatrix} -3.7064 * 0.1591 * 0.938 & -1.4755 * 0.204 * 0.938 & -1.6886 * 0.3685 * 0.938 \\ -3.7064 * 0.1591 * 0.94 & -1.4755 * 0.204 * 0.94 & -1.6886 * 0.3685 * 0.94 \\ -3.7064 * 0.1591 * 0.98 & -1.4755 * 0.204 * 0.98 & -1.6886 * 0.3685 * 0.98 \end{bmatrix}$$

$$= \begin{bmatrix} -0.5531 & -0.0576 & -0.5836 \\ -0.554347 & -0.0577 & -0.5849 \\ -0.577937 & -0.06017 & -0.6098 \end{bmatrix}$$

12

Training – Backpropagation

Hatanın geri yayılımı (Gizli katman 2 ve çıkış katmanı arasındaki ağırlıklar için)

Learning rate = 0,01

$$\hat{W}_{kl} = \begin{bmatrix} W_{k1l1} - (lr * \delta W_{k1l1}) & W_{k1l2} - (lr * \delta W_{k1l2}) & W_{k1l3} - (lr * \delta W_{k1l3}) \\ W_{k2l1} - (lr * \delta W_{k2l1}) & W_{k2l2} - (lr * \delta W_{k2l2}) & W_{k2l3} - (lr * \delta W_{k2l3}) \\ W_{k3l1} - (lr * \delta W_{k3l1}) & W_{k3l2} - (lr * \delta W_{k3l2}) & W_{k3l3} - (lr * \delta W_{k3l3}) \end{bmatrix}$$

$$\hat{W}_{kl} = \begin{bmatrix} 0.1 - (0.01 * -0.5531) & 0.4 - (0.01 * -0.0576) & 0.8 - (0.01 * -0.5836) \\ 0.3 - (0.01 * -0.554347) & 0.7 - (0.01 * -0.0577) & 0.2 - (0.01 * -0.5849) \\ 0.5 - (0.01 * -0.577937) & 0.2 - (0.01 * -0.06017) & 0.9 - (0.01 * -0.6098) \end{bmatrix}$$

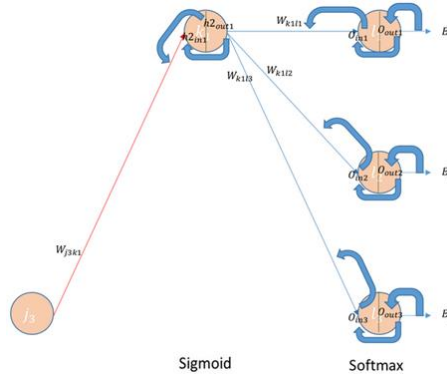
Yeni ağırlık değerleri

$$\hat{W}_{kl} = \begin{bmatrix} 0.105531 & 0.400576 & 0.805836 \\ 0.30055 & 0.700577 & 0.2005849 \\ 0.5005779 & 0.2006017 & 0.9006098 \end{bmatrix}$$

13

Training – Backpropagation

Hatanın geri yayılımı (Gizli katman 1 ve gizli katman 2 arasındaki ağırlıklar için)



$$\frac{\partial h_{2out1}}{\partial h_{2in1}} = \frac{\partial \text{Sigmoid}(h_{2in1})}{\partial h_{2in1}}$$

$$\frac{\partial h_{2out1}}{\partial h_{2in1}} = \text{Sigmoid}(h_{2in1}) * (1 - \text{Sigmoid}(h_{2in1}))$$

$$\frac{\partial h_{2in1}}{\partial W_{j1k1}} = \frac{\partial((h_{1out1} * W_{j1k1}) + (h_{1out2} * W_{j2k1}) + (h_{1out3} * W_{j3k1}) + b_{k1})}{\partial W_{j1k1}} = h_{1out1}$$

14

Training – Backpropagation

Hatanın geri yayılımı (Gizli katman 1 ve gizli katman 2 arasındaki ağırlıklar için)

$$\begin{bmatrix} \frac{\partial h2_{out1}}{\partial h2_{in1}} \\ \frac{\partial h2_{out2}}{\partial h2_{in2}} \\ \frac{\partial h2_{out3}}{\partial h2_{in3}} \end{bmatrix} = \begin{bmatrix} \text{Sigmoid}(h2_{in1}) * (1 - \text{Sigmoid}(h2_{in1})) \\ \text{Sigmoid}(h2_{in1}) * (1 - \text{Sigmoid}(h2_{in1})) \\ \text{Sigmoid}(h2_{in1}) * (1 - \text{Sigmoid}(h2_{in1})) \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial h2_{out1}}{\partial h2_{in1}} \\ \frac{\partial h2_{out2}}{\partial h2_{in2}} \\ \frac{\partial h2_{out3}}{\partial h2_{in3}} \end{bmatrix} = \begin{bmatrix} \text{Sigmoid}(2.73) * (1 - \text{Sigmoid}(2.73)) \\ \text{Sigmoid}(2.76) * (1 - \text{Sigmoid}(2.76)) \\ \text{Sigmoid}(4.001) * (1 - \text{Sigmoid}(4.001)) \end{bmatrix} = \begin{bmatrix} 0.058156 \\ 0.0564 \\ 0.0196 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial h2_{in1}}{\partial W_{j1k1}} \\ \frac{\partial h2_{in1}}{\partial W_{j2k1}} \\ \frac{\partial h2_{in1}}{\partial W_{j3k1}} \end{bmatrix} = \begin{bmatrix} h1_{out1} \\ h1_{out2} \\ h1_{out3} \end{bmatrix} = \begin{bmatrix} 1.35 \\ 1.27 \\ 1.8 \end{bmatrix} \quad \begin{bmatrix} \frac{\partial h2_{in2}}{\partial W_{j1k2}} \\ \frac{\partial h2_{in2}}{\partial W_{j2k2}} \\ \frac{\partial h2_{in2}}{\partial W_{j3k2}} \end{bmatrix} = \begin{bmatrix} h1_{out1} \\ h1_{out2} \\ h1_{out3} \end{bmatrix} = \begin{bmatrix} 1.35 \\ 1.27 \\ 1.8 \end{bmatrix} \quad \begin{bmatrix} \frac{\partial h2_{in3}}{\partial W_{j1k3}} \\ \frac{\partial h2_{in3}}{\partial W_{j2k3}} \\ \frac{\partial h2_{in3}}{\partial W_{j3k3}} \end{bmatrix} = \begin{bmatrix} h1_{out1} \\ h1_{out2} \\ h1_{out3} \end{bmatrix} = \begin{bmatrix} 1.35 \\ 1.27 \\ 1.8 \end{bmatrix}$$

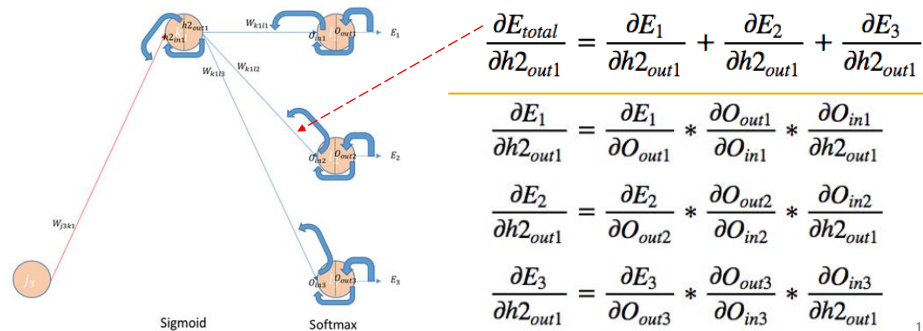
15

Training – Backpropagation

Hatanın geri yayılımı (Gizli katman 1 ve gizli katman 2 arasındaki ağırlıklar için)

$$\frac{\partial E_{total}}{\partial W_{j3k1}} = \frac{\partial E_{total}}{\partial h2_{out1}} * \frac{\partial h2_{out1}}{\partial h2_{in1}} * \frac{\partial h2_{in1}}{\partial W_{j3k1}}$$

$$\delta W_{jk} = \begin{bmatrix} \frac{\partial E_{total}}{\partial W_{j1k1}} & \frac{\partial E_{total}}{\partial W_{j1k2}} & \frac{\partial E_{total}}{\partial W_{j1k3}} \\ \frac{\partial E_{total}}{\partial W_{j2k1}} & \frac{\partial E_{total}}{\partial W_{j2k2}} & \frac{\partial E_{total}}{\partial W_{j2k3}} \\ \frac{\partial E_{total}}{\partial W_{j3k1}} & \frac{\partial E_{total}}{\partial W_{j3k2}} & \frac{\partial E_{total}}{\partial W_{j3k3}} \end{bmatrix} = \begin{bmatrix} \frac{\partial E_{total}}{\partial h2_{out1}} * \frac{\partial h2_{out1}}{\partial h2_{in1}} * \frac{\partial h2_{in1}}{\partial W_{j1k1}} & \frac{\partial E_{total}}{\partial h2_{out2}} * \frac{\partial h2_{out2}}{\partial h2_{in2}} * \frac{\partial h2_{in2}}{\partial W_{j1k2}} & \frac{\partial E_{total}}{\partial h2_{out3}} * \frac{\partial h2_{out3}}{\partial h2_{in3}} * \frac{\partial h2_{in3}}{\partial W_{j1k3}} \\ \frac{\partial E_{total}}{\partial h2_{out1}} * \frac{\partial h2_{out1}}{\partial h2_{in1}} * \frac{\partial h2_{in1}}{\partial W_{j2k1}} & \frac{\partial E_{total}}{\partial h2_{out2}} * \frac{\partial h2_{out2}}{\partial h2_{in2}} * \frac{\partial h2_{in2}}{\partial W_{j2k2}} & \frac{\partial E_{total}}{\partial h2_{out3}} * \frac{\partial h2_{out3}}{\partial h2_{in3}} * \frac{\partial h2_{in3}}{\partial W_{j2k3}} \\ \frac{\partial E_{total}}{\partial h2_{out1}} * \frac{\partial h2_{out1}}{\partial h2_{in1}} * \frac{\partial h2_{in1}}{\partial W_{j3k1}} & \frac{\partial E_{total}}{\partial h2_{out2}} * \frac{\partial h2_{out2}}{\partial h2_{in2}} * \frac{\partial h2_{in2}}{\partial W_{j3k2}} & \frac{\partial E_{total}}{\partial h2_{out3}} * \frac{\partial h2_{out3}}{\partial h2_{in3}} * \frac{\partial h2_{in3}}{\partial W_{j3k3}} \end{bmatrix}$$



16

Training – Backpropagation

Hatanın geri yayılımı (Gizli katman 1 ve gizli katman 2 arasındaki ağırlıklar için)

Sonuç matrisi

$$\begin{bmatrix} \frac{\partial E_{total}}{\partial h_{2out1}} \\ \frac{\partial E_{total}}{\partial h_{2out2}} \\ \frac{\partial E_{total}}{\partial h_{2out3}} \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial E_1}{\partial O_{out1}} * \frac{\partial O_{out1}}{\partial O_{in1}} * \frac{\partial O_{in1}}{\partial h_{2out1}} \right) + \left(\frac{\partial E_2}{\partial O_{out2}} * \frac{\partial O_{out2}}{\partial O_{in2}} * \frac{\partial O_{in2}}{\partial h_{2out1}} \right) + \left(\frac{\partial E_3}{\partial O_{out3}} * \frac{\partial O_{out3}}{\partial O_{in3}} * \frac{\partial O_{in3}}{\partial h_{2out1}} \right) \\ \left(\frac{\partial E_1}{\partial O_{out1}} * \frac{\partial O_{out1}}{\partial O_{in1}} * \frac{\partial O_{in1}}{\partial h_{2out2}} \right) + \left(\frac{\partial E_2}{\partial O_{out2}} * \frac{\partial O_{out2}}{\partial O_{in2}} * \frac{\partial O_{in2}}{\partial h_{2out2}} \right) + \left(\frac{\partial E_3}{\partial O_{out3}} * \frac{\partial O_{out3}}{\partial O_{in3}} * \frac{\partial O_{in3}}{\partial h_{2out2}} \right) \\ \left(\frac{\partial E_1}{\partial O_{out1}} * \frac{\partial O_{out1}}{\partial O_{in1}} * \frac{\partial O_{in1}}{\partial h_{2out3}} \right) + \left(\frac{\partial E_2}{\partial O_{out2}} * \frac{\partial O_{out2}}{\partial O_{in2}} * \frac{\partial O_{in2}}{\partial h_{2out3}} \right) + \left(\frac{\partial E_3}{\partial O_{out3}} * \frac{\partial O_{out3}}{\partial O_{in3}} * \frac{\partial O_{in3}}{\partial h_{2out3}} \right) \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial O_{in1}}{\partial h_{2out1}} & \frac{\partial O_{in2}}{\partial h_{2out1}} & \frac{\partial O_{in3}}{\partial h_{2out1}} \\ \frac{\partial O_{in1}}{\partial h_{2out2}} & \frac{\partial O_{in2}}{\partial h_{2out2}} & \frac{\partial O_{in3}}{\partial h_{2out2}} \\ \frac{\partial O_{in1}}{\partial h_{2out3}} & \frac{\partial O_{in2}}{\partial h_{2out3}} & \frac{\partial O_{in3}}{\partial h_{2out3}} \end{bmatrix} = \begin{bmatrix} W_{k111} & W_{k112} & W_{k113} \\ W_{k211} & W_{k212} & W_{k213} \\ W_{k311} & W_{k312} & W_{k313} \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial E_{total}}{\partial h_{2out1}} \\ \frac{\partial E_{total}}{\partial h_{2out2}} \\ \frac{\partial E_{total}}{\partial h_{2out3}} \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial E_1}{\partial O_{out1}} * \frac{\partial O_{out1}}{\partial O_{in1}} * W_{k111} \right) + \left(\frac{\partial E_2}{\partial O_{out2}} * \frac{\partial O_{out2}}{\partial O_{in2}} * W_{k112} \right) + \left(\frac{\partial E_3}{\partial O_{out3}} * \frac{\partial O_{out3}}{\partial O_{in3}} * W_{k113} \right) \\ \left(\frac{\partial E_1}{\partial O_{out1}} * \frac{\partial O_{out1}}{\partial O_{in1}} * W_{k211} \right) + \left(\frac{\partial E_2}{\partial O_{out2}} * \frac{\partial O_{out2}}{\partial O_{in2}} * W_{k212} \right) + \left(\frac{\partial E_3}{\partial O_{out3}} * \frac{\partial O_{out3}}{\partial O_{in3}} * W_{k213} \right) \\ \left(\frac{\partial E_1}{\partial O_{out1}} * \frac{\partial O_{out1}}{\partial O_{in1}} * W_{k311} \right) + \left(\frac{\partial E_2}{\partial O_{out2}} * \frac{\partial O_{out2}}{\partial O_{in2}} * W_{k312} \right) + \left(\frac{\partial E_3}{\partial O_{out3}} * \frac{\partial O_{out3}}{\partial O_{in3}} * W_{k313} \right) \end{bmatrix}$$

17

Training – Backpropagation

Hatanın geri yayılımı (Gizli katman 1 ve gizli katman 2 arasındaki ağırlıklar için)

$$\begin{bmatrix} \frac{\partial E_{total}}{\partial h_{2out1}} \\ \frac{\partial E_{total}}{\partial h_{2out2}} \\ \frac{\partial E_{total}}{\partial h_{2out3}} \end{bmatrix} = \begin{bmatrix} (-3.70644 * 0.15911 * 0.1) + (-1.4755 * 0.2040 * 0.4) + (-1.6886 * 0.3685 * 0.8) \\ (-3.70644 * 0.15911 * 0.3) + (-1.4755 * 0.2040 * 0.7) + (-1.6886 * 0.3685 * 0.2) \\ (-3.70644 * 0.15911 * 0.5) + (-1.4755 * 0.2040 * 0.2) + (-1.6886 * 0.3685 * 0.9) \end{bmatrix}$$

$$= \begin{bmatrix} (-0.0589) + (-0.2383) + (-0.5931) \\ (-0.1769) + (-0.417) + (-0.14828) \\ (-0.2948) + (-0.119) + (-0.667) \end{bmatrix} = \begin{bmatrix} -0.8903 \\ -0.74218 \\ -1.0810 \end{bmatrix}$$

$$\delta W_{jk} = \begin{bmatrix} \frac{\partial E_{total}}{\partial W_{j1k1}} & \frac{\partial E_{total}}{\partial W_{j1k2}} & \frac{\partial E_{total}}{\partial W_{j1k3}} \\ \frac{\partial E_{total}}{\partial W_{j2k1}} & \frac{\partial E_{total}}{\partial W_{j2k2}} & \frac{\partial E_{total}}{\partial W_{j2k3}} \\ \frac{\partial E_{total}}{\partial W_{j3k1}} & \frac{\partial E_{total}}{\partial W_{j3k2}} & \frac{\partial E_{total}}{\partial W_{j3k3}} \end{bmatrix} = \begin{bmatrix} \frac{\partial E_{total}}{\partial h_{2out1}} * \frac{\partial h_{2out1}}{\partial h_{2in1}} * \frac{\partial h_{2in1}}{\partial W_{j1k1}} & \frac{\partial E_{total}}{\partial h_{2out2}} * \frac{\partial h_{2out2}}{\partial h_{2in2}} * \frac{\partial h_{2in2}}{\partial W_{j1k2}} & \frac{\partial E_{total}}{\partial h_{2out3}} * \frac{\partial h_{2out3}}{\partial h_{2in3}} * \frac{\partial h_{2in3}}{\partial W_{j1k3}} \\ \frac{\partial E_{total}}{\partial h_{2out1}} * \frac{\partial h_{2out1}}{\partial h_{2in1}} * \frac{\partial h_{2in1}}{\partial W_{j2k1}} & \frac{\partial E_{total}}{\partial h_{2out2}} * \frac{\partial h_{2out2}}{\partial h_{2in2}} * \frac{\partial h_{2in2}}{\partial W_{j2k2}} & \frac{\partial E_{total}}{\partial h_{2out3}} * \frac{\partial h_{2out3}}{\partial h_{2in3}} * \frac{\partial h_{2in3}}{\partial W_{j2k3}} \\ \frac{\partial E_{total}}{\partial h_{2out1}} * \frac{\partial h_{2out1}}{\partial h_{2in1}} * \frac{\partial h_{2in1}}{\partial W_{j3k1}} & \frac{\partial E_{total}}{\partial h_{2out2}} * \frac{\partial h_{2out2}}{\partial h_{2in2}} * \frac{\partial h_{2in2}}{\partial W_{j3k2}} & \frac{\partial E_{total}}{\partial h_{2out3}} * \frac{\partial h_{2out3}}{\partial h_{2in3}} * \frac{\partial h_{2in3}}{\partial W_{j3k3}} \end{bmatrix}$$

$$\delta W_{jk} = \begin{bmatrix} -0.8903 * 0.058156 * 1.35 & -0.74218 * 0.0564 * 1.35 & -1.0810 * 0.0196 * 1.35 \\ -0.8903 * 0.058156 * 1.27 & -0.74218 * 0.0564 * 1.27 & -1.0810 * 0.0196 * 1.27 \\ -0.8903 * 0.058156 * 1.8 & -0.74218 * 0.0564 * 1.8 & -1.0810 * 0.0196 * 1.8 \end{bmatrix}$$

$$\delta W_{jk} = \begin{bmatrix} -0.06989 & -0.0565 & -0.0286 \\ -0.06575 & -0.05316 & -0.0269 \\ -0.0932 & -0.0753 & -0.03813 \end{bmatrix}$$

18

Training – Backpropagation

Hatanın geri yayılımı (Gizli katman 1 ve gizli katman 2 arasındaki ağırlıklar için)

Yeni ağırlık değerleri

$$\dot{W}_{jk} = \begin{bmatrix} W_{j1k1} - (lr * \delta W_{j1k1}) & W_{j1k2} - (lr * \delta W_{j1k2}) & W_{j1k3} - (lr * \delta W_{j1k3}) \\ W_{j2k1} - (lr * \delta W_{j2k1}) & W_{j2k2} - (lr * \delta W_{j2k2}) & W_{j2k3} - (lr * \delta W_{j2k3}) \\ W_{j3k1} - (lr * \delta W_{j3k1}) & W_{j3k2} - (lr * \delta W_{j3k2}) & W_{j3k3} - (lr * \delta W_{j3k3}) \end{bmatrix}$$

$$\dot{W}_{jk} = \begin{bmatrix} 0.2 - (0.01 * -0.06989) & 0.3 - (0.01 * -0.0565) & 0.5 - (0.01 * -0.0286) \\ 0.3 - (0.01 * -0.06575) & 0.5 - (0.01 * -0.05316) & 0.7 - (0.01 * -0.0269) \\ 0.6 - (0.01 * -0.0932) & 0.4 - (0.01 * -0.0753) & 0.8 - (0.01 * -0.03813) \end{bmatrix}$$

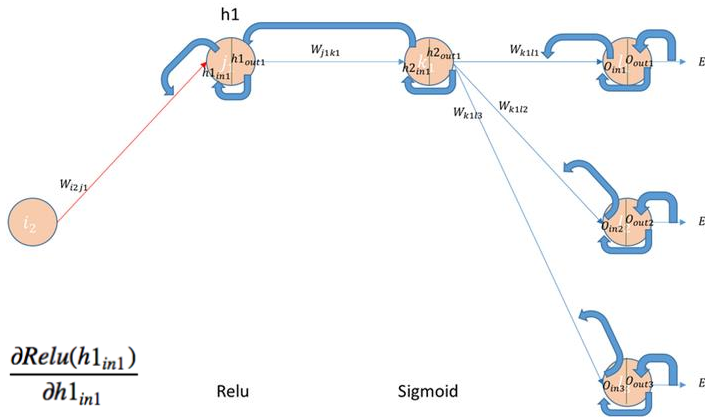
Yeni ağırlık değerleri

$$\dot{W}_{jk} = \begin{bmatrix} 0.2006989 & 0.300565 & 0.500286 \\ 0.3006575 & 0.5005316 & 0.700269 \\ 0.600932 & 0.400753 & 0.803813 \end{bmatrix}$$

19

Training – Backpropagation

Hatanın geri yayılımı (Giriş katmanı ve gizli katman 1 arasındaki ağırlıklar için)



$$\frac{\partial h1_{out1}}{\partial h1_{in1}} = \frac{\partial Relu(h1_{in1})}{\partial h1_{in1}}$$

Relu

Sigmoid

$$\frac{\partial h1_{in1}}{\partial W_{ilj1}} = \frac{\partial((I_{out1} * W_{i1j1}) + (I_{out2} * W_{i2j1}) + (I_{out3} * W_{i3j1}) + b_{j1})}{\partial W_{ilj1}}$$

$$= I_{out1}$$

20

Training – Backpropagation

Hatanın geri yayılımı (Giriş katmanı ve gizli katman 1 arasındaki ağırlıklar için)

Tüm girişler pozitif olduğundan hepsi 1 olur.

$$\frac{\partial h_{out1}}{\partial h_{in1}} = 1.0$$

$$\begin{bmatrix} \frac{\partial h_{out1}}{\partial h_{in1}} \\ \frac{\partial h_{out2}}{\partial h_{in2}} \\ \frac{\partial h_{out3}}{\partial h_{in3}} \end{bmatrix} = \begin{bmatrix} 1.0 \\ 1.0 \\ 1.0 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial h_{in1}}{\partial W_{1j1}} \\ \frac{\partial h_{in1}}{\partial W_{2j1}} \\ \frac{\partial h_{in1}}{\partial W_{3j1}} \end{bmatrix} = \begin{bmatrix} I_{out1} \\ I_{out2} \\ I_{out3} \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.7 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial h_{in2}}{\partial W_{1j2}} \\ \frac{\partial h_{in2}}{\partial W_{2j2}} \\ \frac{\partial h_{in2}}{\partial W_{3j2}} \end{bmatrix} = \begin{bmatrix} I_{out1} \\ I_{out2} \\ I_{out3} \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.7 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial h_{in3}}{\partial W_{1j3}} \\ \frac{\partial h_{in3}}{\partial W_{2j3}} \\ \frac{\partial h_{in3}}{\partial W_{3j3}} \end{bmatrix} = \begin{bmatrix} I_{out1} \\ I_{out2} \\ I_{out3} \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.7 \end{bmatrix}$$

$$\frac{\partial E_{total}}{\partial W_{i2j1}} = \frac{\partial E_{total}}{\partial h_{out1}} * \frac{\partial h_{out1}}{\partial h_{in1}} * \frac{\partial h_{in1}}{\partial W_{i2j1}}$$

21

Training – Backpropagation

Hatanın geri yayılımı (Giriş katmanı ve gizli katman 1 arasındaki ağırlıklar için)

$$\delta W_{ij} = \begin{bmatrix} \frac{\partial E_{total}}{\partial W_{1j1}} & \frac{\partial E_{total}}{\partial W_{1j2}} & \frac{\partial E_{total}}{\partial W_{1j3}} \\ \frac{\partial E_{total}}{\partial W_{2j1}} & \frac{\partial E_{total}}{\partial W_{2j2}} & \frac{\partial E_{total}}{\partial W_{2j3}} \\ \frac{\partial E_{total}}{\partial W_{3j1}} & \frac{\partial E_{total}}{\partial W_{3j2}} & \frac{\partial E_{total}}{\partial W_{3j3}} \end{bmatrix} = \begin{bmatrix} \frac{\partial E_{total}}{\partial h_{out1}} * \frac{\partial h_{out1}}{\partial h_{in1}} * \frac{\partial h_{in1}}{\partial W_{1j1}} & \frac{\partial E_{total}}{\partial h_{out1}} * \frac{\partial h_{out1}}{\partial h_{in1}} * \frac{\partial h_{in1}}{\partial W_{1j2}} & \frac{\partial E_{total}}{\partial h_{out1}} * \frac{\partial h_{out1}}{\partial h_{in1}} * \frac{\partial h_{in1}}{\partial W_{1j3}} \\ \frac{\partial E_{total}}{\partial h_{out1}} * \frac{\partial h_{out1}}{\partial h_{in1}} * \frac{\partial h_{in1}}{\partial W_{2j1}} & \frac{\partial E_{total}}{\partial h_{out1}} * \frac{\partial h_{out1}}{\partial h_{in1}} * \frac{\partial h_{in1}}{\partial W_{2j2}} & \frac{\partial E_{total}}{\partial h_{out1}} * \frac{\partial h_{out1}}{\partial h_{in1}} * \frac{\partial h_{in1}}{\partial W_{2j3}} \\ \frac{\partial E_{total}}{\partial h_{out1}} * \frac{\partial h_{out1}}{\partial h_{in1}} * \frac{\partial h_{in1}}{\partial W_{3j1}} & \frac{\partial E_{total}}{\partial h_{out1}} * \frac{\partial h_{out1}}{\partial h_{in1}} * \frac{\partial h_{in1}}{\partial W_{3j2}} & \frac{\partial E_{total}}{\partial h_{out1}} * \frac{\partial h_{out1}}{\partial h_{in1}} * \frac{\partial h_{in1}}{\partial W_{3j3}} \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial E_{total}}{\partial h_{out1}} \\ \frac{\partial E_{total}}{\partial h_{out2}} \\ \frac{\partial E_{total}}{\partial h_{out3}} \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial E_{total}}{\partial h_{2out1}} * \frac{\partial h_{2out1}}{\partial h_{2in1}} * \frac{\partial h_{2in1}}{\partial h_{1out1}} \right) \\ \left(\frac{\partial E_{total}}{\partial h_{2out2}} * \frac{\partial h_{2out2}}{\partial h_{2in2}} * \frac{\partial h_{2in2}}{\partial h_{1out2}} \right) \\ \left(\frac{\partial E_{total}}{\partial h_{2out3}} * \frac{\partial h_{2out3}}{\partial h_{2in3}} * \frac{\partial h_{2in3}}{\partial h_{1out3}} \right) \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial E_{total}}{\partial h_{out1}} \\ \frac{\partial E_{total}}{\partial h_{out2}} \\ \frac{\partial E_{total}}{\partial h_{out3}} \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial E_{total}}{\partial h_{2out1}} * \frac{\partial h_{2out1}}{\partial h_{2in1}} * W_{j1k1} \right) \\ \left(\frac{\partial E_{total}}{\partial h_{2out2}} * \frac{\partial h_{2out2}}{\partial h_{2in2}} * W_{j2k2} \right) \\ \left(\frac{\partial E_{total}}{\partial h_{2out3}} * \frac{\partial h_{2out3}}{\partial h_{2in3}} * W_{j3k3} \right) \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial E_{total}}{\partial h_{out1}} \\ \frac{\partial E_{total}}{\partial h_{out2}} \\ \frac{\partial E_{total}}{\partial h_{out3}} \end{bmatrix} = \begin{bmatrix} -0.8903 * 0.058156 * 0.2 \\ -0.74218 * 0.0564 * 0.5 \\ -1.0810 * 0.0196 * 0.8 \end{bmatrix} = \begin{bmatrix} -0.01035 \\ -0.0209 \\ -0.0169 \end{bmatrix}$$

22

Training – Backpropagation

Hatanın geri yayılımı (Giriş katmanı ve gizli katman 1 arasındaki ağırlıklar için)

$$\delta W_{ij} = \begin{bmatrix} \frac{\partial E_{total}}{\partial W_{i1j1}} & \frac{\partial E_{total}}{\partial W_{i1j2}} & \frac{\partial E_{total}}{\partial W_{i1j3}} \\ \frac{\partial E_{total}}{\partial W_{i2j1}} & \frac{\partial E_{total}}{\partial W_{i2j2}} & \frac{\partial E_{total}}{\partial W_{i2j3}} \\ \frac{\partial E_{total}}{\partial W_{i3j1}} & \frac{\partial E_{total}}{\partial W_{i3j2}} & \frac{\partial E_{total}}{\partial W_{i3j3}} \end{bmatrix} = \begin{bmatrix} \frac{\partial E_{total}}{\partial h_{1out1}} * \frac{\partial h_{1out1}}{\partial h_{1in1}} * \frac{\partial h_{1in1}}{\partial W_{i1j1}} & \frac{\partial E_{total}}{\partial h_{1out1}} * \frac{\partial h_{1out1}}{\partial h_{1in2}} * \frac{\partial h_{1in2}}{\partial W_{i1j2}} & \frac{\partial E_{total}}{\partial h_{1out1}} * \frac{\partial h_{1out1}}{\partial h_{1in3}} * \frac{\partial h_{1in3}}{\partial W_{i1j3}} \\ \frac{\partial E_{total}}{\partial h_{1out2}} * \frac{\partial h_{1out2}}{\partial h_{1in1}} * \frac{\partial h_{1in1}}{\partial W_{i2j1}} & \frac{\partial E_{total}}{\partial h_{1out2}} * \frac{\partial h_{1out2}}{\partial h_{1in2}} * \frac{\partial h_{1in2}}{\partial W_{i2j2}} & \frac{\partial E_{total}}{\partial h_{1out2}} * \frac{\partial h_{1out2}}{\partial h_{1in3}} * \frac{\partial h_{1in3}}{\partial W_{i2j3}} \\ \frac{\partial E_{total}}{\partial h_{1out3}} * \frac{\partial h_{1out3}}{\partial h_{1in1}} * \frac{\partial h_{1in1}}{\partial W_{i3j1}} & \frac{\partial E_{total}}{\partial h_{1out3}} * \frac{\partial h_{1out3}}{\partial h_{1in2}} * \frac{\partial h_{1in2}}{\partial W_{i3j2}} & \frac{\partial E_{total}}{\partial h_{1out3}} * \frac{\partial h_{1out3}}{\partial h_{1in3}} * \frac{\partial h_{1in3}}{\partial W_{i3j3}} \end{bmatrix}$$

$$\delta W_{ij} = \begin{bmatrix} -0.01035 * 1 * 0.1 & -0.0209 * 1 * 0.1 & -0.0169 * 1 * 0.1 \\ -0.01035 * 1 * 0.2 & -0.0209 * 1 * 0.2 & -0.0169 * 1 * 0.2 \\ -0.01035 * 1 * 0.7 & -0.0209 * 1 * 0.7 & -0.0169 * 1 * 0.7 \end{bmatrix} = \begin{bmatrix} -0.001035 & -0.00209 & -0.00169 \\ -0.00207 & -0.00418 & -0.00338 \\ -0.007245 & -0.01463 & -0.01183 \end{bmatrix}$$

$$\hat{W}_{ij} = \begin{bmatrix} W_{i1j1} - (lr * \delta W_{i1j1}) & W_{i1j2} - (lr * \delta W_{i1j2}) & W_{i1j3} - (lr * \delta W_{i1j3}) \\ W_{i2j1} - (lr * \delta W_{i2j1}) & W_{i2j2} - (lr * \delta W_{i2j2}) & W_{i2j3} - (lr * \delta W_{i2j3}) \\ W_{i3j1} - (lr * \delta W_{i3j1}) & W_{i3j2} - (lr * \delta W_{i3j2}) & W_{i3j3} - (lr * \delta W_{i3j3}) \end{bmatrix}$$

$$\hat{W}_{ij} = \begin{bmatrix} 0.1 - (0.01 * -0.001035) & 0.2 - (0.01 * -0.00209) & 0.3 - (0.01 * -0.00169) \\ 0.3 - (0.01 * -0.00207) & 0.2 - (0.01 * -0.00418) & 0.7 - (0.01 * -0.00338) \\ 0.4 - (0.01 * -0.007245) & 0.3 - (0.01 * -0.01463) & 0.9 - (0.01 * -0.01183) \end{bmatrix}$$

Yeni ağırlık değerleri

$$\hat{W}_{ij} = \begin{bmatrix} 0.10001035 & 0.2000209 & 0.3000169 \\ 0.3000207 & 0.2000418 & 0.7000338 \\ 0.40007245 & 0.3001463 & 0.9001183 \end{bmatrix}$$

23

Training – Backpropagation

Tüm ağırlıkların yeni değerleri

Ağırlıkların başlangıç değerleri

$$W_{ij} = \begin{bmatrix} W_{i1j1} & W_{i1j2} & W_{i1j3} \\ W_{i2j1} & W_{i2j2} & W_{i2j3} \\ W_{i3j1} & W_{i3j2} & W_{i3j3} \end{bmatrix} = \begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0.3 & 0.2 & 0.7 \\ 0.4 & 0.3 & 0.9 \end{bmatrix}$$

$$W_{jk} = \begin{bmatrix} W_{j1k1} & W_{j1k2} & W_{j1k3} \\ W_{j2k1} & W_{j2k2} & W_{j2k3} \\ W_{j3k1} & W_{j3k2} & W_{j3k3} \end{bmatrix} = \begin{bmatrix} 0.2 & 0.3 & 0.5 \\ 0.3 & 0.5 & 0.7 \\ 0.6 & 0.4 & 0.8 \end{bmatrix}$$

$$W_{kl} = \begin{bmatrix} W_{k1l1} & W_{k1l2} & W_{k1l3} \\ W_{k2l1} & W_{k2l2} & W_{k2l3} \\ W_{k3l1} & W_{k3l2} & W_{k3l3} \end{bmatrix} = \begin{bmatrix} 0.1 & 0.4 & 0.8 \\ 0.3 & 0.7 & 0.2 \\ 0.5 & 0.2 & 0.9 \end{bmatrix}$$

Ağırlıkların yeni değerleri

$$\hat{W}_{ij} = \begin{bmatrix} 0.10001035 & 0.2000209 & 0.3000169 \\ 0.3000207 & 0.2000418 & 0.7000338 \\ 0.40007245 & 0.3001463 & 0.9001183 \end{bmatrix}$$

$$\hat{W}_{jk} = \begin{bmatrix} 0.2006989 & 0.300565 & 0.500286 \\ 0.3006575 & 0.5005316 & 0.700269 \\ 0.600932 & 0.400753 & 0.803813 \end{bmatrix}$$

$$\hat{W}_{kl} = \begin{bmatrix} 0.105531 & 0.400576 & 0.805836 \\ 0.30055 & 0.700577 & 0.2005849 \\ 0.5005779 & 0.2006017 & 0.9006098 \end{bmatrix}$$

24

Training – Backpropagation

FC katmanı

- ▶ FC katmanında klasik ANN ağırlıkları gibi eğitim yapılır.
- ▶ **Giriş ile ağırlıkların çarpıldığı** LeNet mimarisinde geriye yayılım için çarpma işlemi yapılır.

İleri geçiş

$$y_l = w_l * y_{l-1}$$

Ardışık katman çıkışları arasında geriye geçiş

$$\frac{\partial E}{\partial y_{l-1}} = \frac{\partial E}{\partial y_l} \frac{\partial y_l}{\partial y_{l-1}} = \frac{\partial E}{\partial y_l} w_l$$

İlgili ağırlık için değişim değeri

$$\delta w_l = \frac{\partial E}{\partial w_l} = \frac{\partial E}{\partial y_l} \frac{\partial y_l}{\partial w_l} = \frac{\partial E}{\partial y_l} y_{l-1}$$

- ▶ İleriye doğru geçişte y_{l-1} (önceki katman çıkışı) değeri saklanır.

25

Training – Backpropagation

Convolution katmanı

- ▶ Çıktadaki hataya göre filtredeki ağırlık (piksel) değerleri değişir.
- ▶ Şekilde **O** çıkış matrisini, **X** giriş matrisini ve **F** filtre matrisini göstermektedir.
- ▶ İleriye doğru geçişte filtre kaydırılarak çıkış değerleri hesaplanır.

$$\text{Convolution} \left(\begin{array}{|c|c|c|} \hline X_{11} & X_{12} & X_{13} \\ \hline X_{21} & X_{22} & X_{23} \\ \hline X_{31} & X_{32} & X_{33} \\ \hline \end{array} , \begin{array}{|c|c|} \hline F_{11} & F_{12} \\ \hline F_{21} & F_{22} \\ \hline \end{array} \right) = \begin{array}{|c|c|} \hline O_{11} & O_{12} \\ \hline O_{21} & O_{22} \\ \hline \end{array}$$

$$O_{11} = F_{11}X_{11} + F_{12}X_{12} + F_{21}X_{21} + F_{22}X_{22}$$

$$O_{12} = F_{11}X_{12} + F_{12}X_{13} + F_{21}X_{22} + F_{22}X_{23}$$

$$O_{21} = F_{11}X_{21} + F_{12}X_{22} + F_{21}X_{31} + F_{22}X_{32}$$

$$O_{22} = F_{11}X_{22} + F_{12}X_{23} + F_{21}X_{32} + F_{22}X_{33}$$

26

Training – Backpropagation

Convolution katmanı

- Geriye yayılımda hatanın filtre matrisindeki değerlere göre kısmi türevi alınır.

$$\frac{\partial E}{\partial F_{11}} = \frac{\partial E}{\partial O_{11}} \frac{\partial O_{11}}{\partial F_{11}} + \frac{\partial E}{\partial O_{12}} \frac{\partial O_{12}}{\partial F_{11}} + \frac{\partial E}{\partial O_{21}} \frac{\partial O_{21}}{\partial F_{11}} + \frac{\partial E}{\partial O_{22}} \frac{\partial O_{22}}{\partial F_{11}}$$

$$\frac{\partial E}{\partial F_{12}} = \frac{\partial E}{\partial O_{11}} \frac{\partial O_{11}}{\partial F_{12}} + \frac{\partial E}{\partial O_{12}} \frac{\partial O_{12}}{\partial F_{12}} + \frac{\partial E}{\partial O_{21}} \frac{\partial O_{21}}{\partial F_{12}} + \frac{\partial E}{\partial O_{22}} \frac{\partial O_{22}}{\partial F_{12}}$$

$$\frac{\partial E}{\partial F_{21}} = \frac{\partial E}{\partial O_{11}} \frac{\partial O_{11}}{\partial F_{21}} + \frac{\partial E}{\partial O_{12}} \frac{\partial O_{12}}{\partial F_{21}} + \frac{\partial E}{\partial O_{21}} \frac{\partial O_{21}}{\partial F_{21}} + \frac{\partial E}{\partial O_{22}} \frac{\partial O_{22}}{\partial F_{21}}$$

$$\frac{\partial E}{\partial F_{22}} = \frac{\partial E}{\partial O_{11}} \frac{\partial O_{11}}{\partial F_{22}} + \frac{\partial E}{\partial O_{12}} \frac{\partial O_{12}}{\partial F_{22}} + \frac{\partial E}{\partial O_{21}} \frac{\partial O_{21}}{\partial F_{22}} + \frac{\partial E}{\partial O_{22}} \frac{\partial O_{22}}{\partial F_{22}}$$

$$\frac{\partial E}{\partial F_{11}} = \frac{\partial E}{\partial O_{11}} X_{11} + \frac{\partial E}{\partial O_{12}} X_{12} + \frac{\partial E}{\partial O_{21}} X_{21} + \frac{\partial E}{\partial O_{22}} X_{22}$$

$$\frac{\partial E}{\partial F_{12}} = \frac{\partial E}{\partial O_{11}} X_{12} + \frac{\partial E}{\partial O_{12}} X_{13} + \frac{\partial E}{\partial O_{21}} X_{22} + \frac{\partial E}{\partial O_{22}} X_{23}$$

$$\frac{\partial E}{\partial F_{21}} = \frac{\partial E}{\partial O_{11}} X_{21} + \frac{\partial E}{\partial O_{12}} X_{22} + \frac{\partial E}{\partial O_{21}} X_{31} + \frac{\partial E}{\partial O_{22}} X_{32}$$

$$\frac{\partial E}{\partial F_{22}} = \frac{\partial E}{\partial O_{11}} X_{22} + \frac{\partial E}{\partial O_{12}} X_{23} + \frac{\partial E}{\partial O_{21}} X_{32} + \frac{\partial E}{\partial O_{22}} X_{33}$$

$$O_{11} = F_{11}X_{11} + F_{12}X_{12} + F_{21}X_{21} + F_{22}X_{22}$$

$$O_{12} = F_{11}X_{12} + F_{12}X_{13} + F_{21}X_{22} + F_{22}X_{23}$$

$$O_{21} = F_{11}X_{21} + F_{12}X_{22} + F_{21}X_{31} + F_{22}X_{32}$$

$$O_{22} = F_{11}X_{22} + F_{12}X_{23} + F_{21}X_{32} + F_{22}X_{33}$$

27

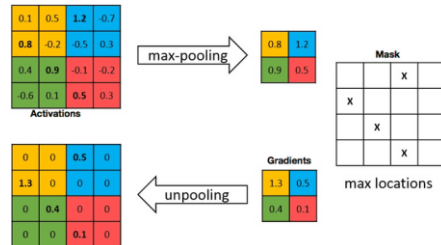
Training – Backpropagation

Pooling katmanı

- Max pooling yapılırsa, hata değeri sadece maksimum olana doğrudan aktarılır ($window\ size = p * q$).
- Average pooling yapılırsa, hata değeri window size'a bölünerek hepsine eşit aktarılır.

Max pooling

$$\frac{\partial L}{\partial y_{l-1}}(x+p, y+q) = \begin{cases} 0, & \text{if } (y_l(x, y)) \neq (y_{l-1}(x+p, y+q)) \\ \frac{\partial L}{\partial y_l}, & \text{otherwise} \end{cases}$$



Average pooling

$$\frac{\partial L}{\partial y_{l-1}}(x+p, y+q) = \frac{1}{p * q} \frac{\partial L}{\partial y_l}$$

28

İçerik

- ▶ Training
- ▶ Değerlendirme
- ▶ CNN mimarileri
- ▶ CNN uygulamaları

29

Değerlendirme

- ▶ Bir **CNN modeli, özellik çıkarımı ve sınıflandırıcının birleşimi** olarak düşünülebilir.
- ▶ **Convolution+pooling** katmanları **özellik çıkarımı yapar** (iki göz, uzun kulaklar, dört ayaklı, tekerlek, vb.).
- ▶ **Fully connected katmanı** bir **sınıflandırıcıdır** ve girişteki görüntünün kedi, köpek, otomobil, vb. olma olasılığını hesaplar.
- ▶ **CNN modeli**, girilen **bir görüntü** ve **atanan bir etiket için** otomatik olarak **özellik algılaması gerçekleştirir.**

30

İçerik

- ▶ Training
- ▶ Değerlendirme
- ▶ CNN mimarileri
- ▶ CNN uygulamaları

CNN mimarileri

- ▶ Yaygın kullanılan CNN mimarileri:
 - ▶ LeNet-5
 - ▶ AlexNet
 - ▶ VGG-16
 - ▶ ResNet
 - ▶ GoogLeNet

CNN mimarileri

LeNet-5

- ▶ 1998 yılında yayınlanmış olan ve **ilk başarılı sonucu veren CNN modelidir.**
- ▶ Banka çekleri üzerindeki **sayıların okunması için geliştirilmiştir.**
- ▶ **MNIST*** veriseti üzerinde başarılı bir şekilde uygulanmıştır.



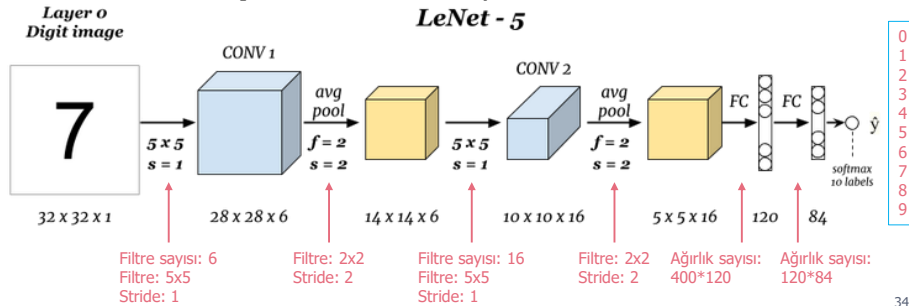
*Modified National Institute of Standards and Technology

33

CNN mimarileri

LeNet-5

- ▶ **LeNet-5 modelinde,** boyut azaltma adımlarında **average pooling** yapılmaktadır.
- ▶ **Aktivasyon fonksiyonu** olarak **sigmoid** ve **hiperbolik tanjant** kullanılmaktadır.
- ▶ FC katmanına $16 \times 5 \times 5 = 400$ parametre girer.
- ▶ Çıkışta 0–9 arasındaki rakamlar için 10 sınıflı softmax vardır.
- ▶ LeNet-5 **60 bin parametre** hesaplamaktadır.



34

CNN mimarileri

AlexNet

- ▶ **CNN modellerinin** ve derin öğrenmenin **popüler hale gelmesini sağlayan ilk çalışmadır.**
- ▶ Alex Krizhevsky, Ilya Sutskever and Geoffrey Hinton tarafından 2012 yılında geliştirilmiştir.
- ▶ **Art arda convolution ve pooling katmanları** olmasından dolayı LeNet modeline benzemektedir.
- ▶ **Aktivasyon fonksiyonu** olarak **ReLU** kullanılmaktadır.
- ▶ Pooling katmanlarında **max-pooling** kullanılmaktadır.

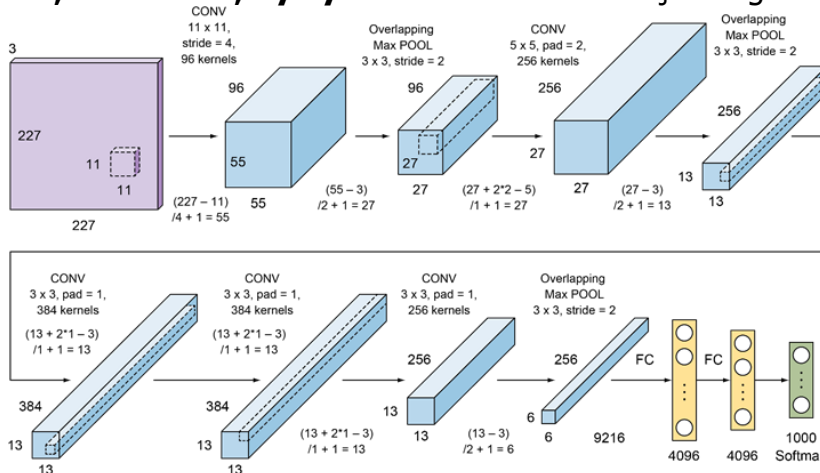
35

CNN mimarileri

AlexNet

*ImageNet Large Scale Visual Recognition Competition

- ▶ Yaklaşık **60 milyon parametre** hesaplanmaktadır.
- ▶ **ILSVRC*** yarışmasında sınıflandırma **doğruluk oranını %74,3'ten %83,6'ya yükselterek** önemli başarı sağlamıştır.



36

CNN mimarileri

VGG-16

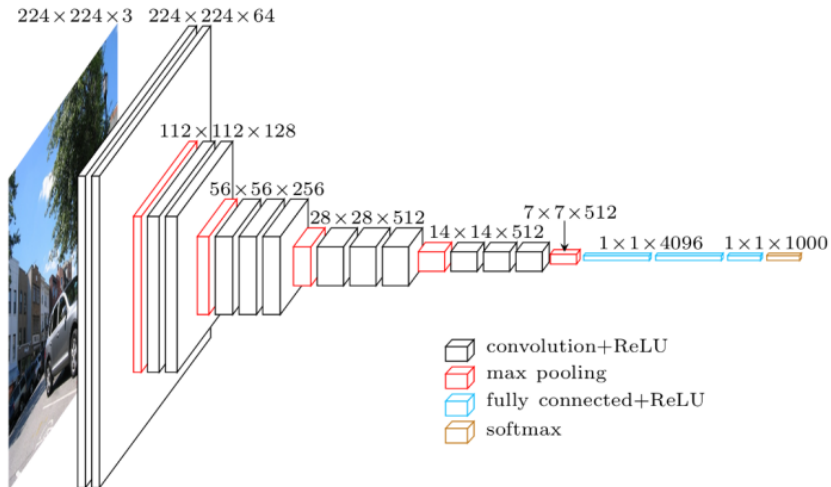
- ▶ VGG-16 modeli **convolution katmanlarını 2'li ya da 3'lü olarak kullanılır.**
- ▶ **FC katmanında $7 \times 7 \times 512 = 4096$ öznitelik vektörü giriş alınır.**
- ▶ **FC katmanı çıkışında 1000 sınıflı softmax sonucu hesaplanır.**
- ▶ Yaklaşık **138 milyon parametre** hesabı yapılmaktadır.

37

CNN mimarileri

VGG-16

- ▶ Girişten **çıkışa doğru** matrislerin yükseklik ve genişlikleri azalırken **derinlik değeri (kanal sayısı) artmaktadır.**

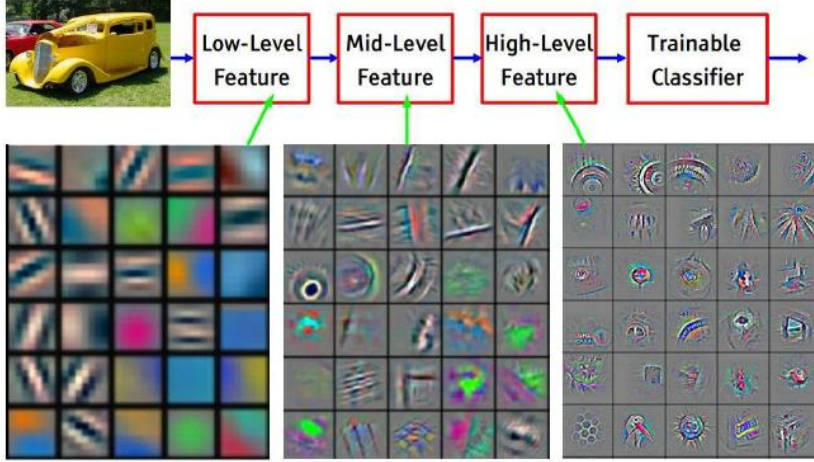


38

CNN mimarileri

VGG-16

- Her konvolüsyon katmanında farklı filtreler kullanılarak görüntünün öznitelikleri elde edilir.



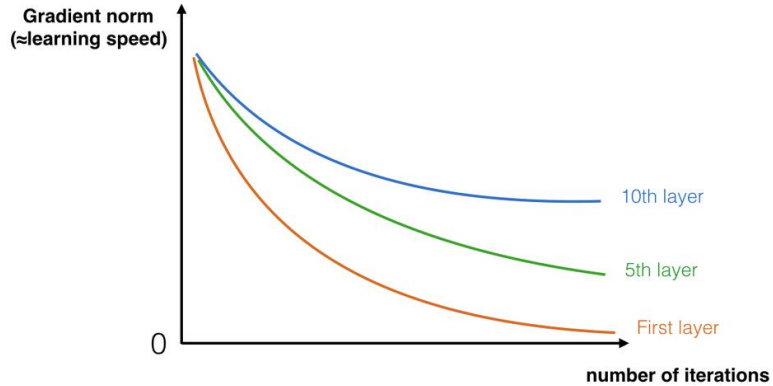
Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

39

CNN mimarileri

ResNet

- Eğitim sürecinde **ilk katmanlara gelirken öğrenme hızı** önemli oranda **düşer** (vanishing gradient).

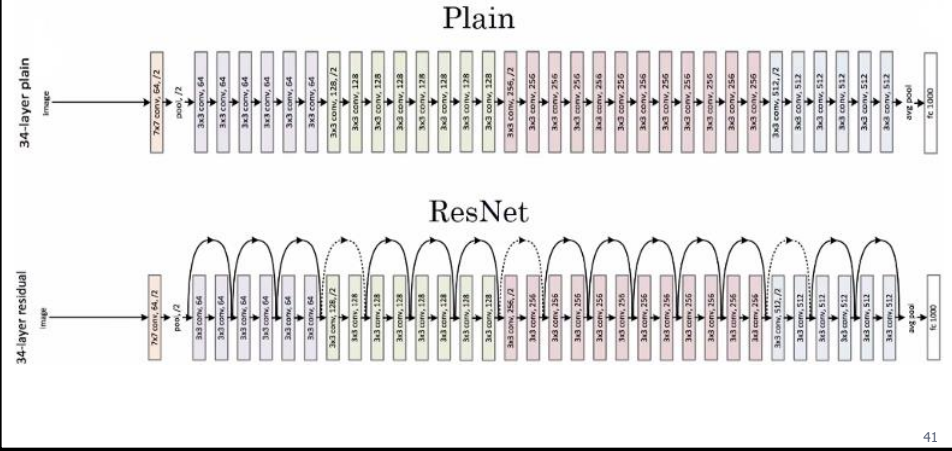


40

CNN mimarileri

ResNet

- ▶ ResNet (Residual Net) modelinde, **ardışık olmayan katmanlar arasında veri aktarılır.**
- ▶ Katman çıkışı **iki sonraki katman girişine eklenir.**

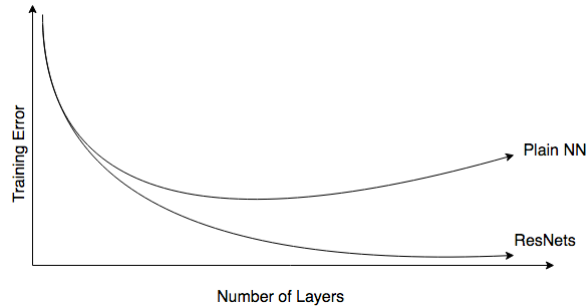


41

CNN mimarileri

ResNet

- ▶ ResNet modelinde, **eğitim hatası katman sayısı arttıkça azalır.**

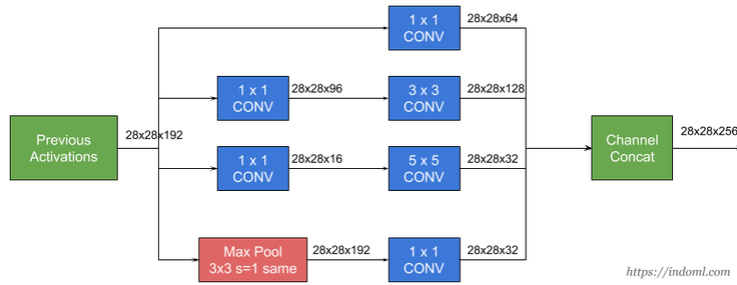
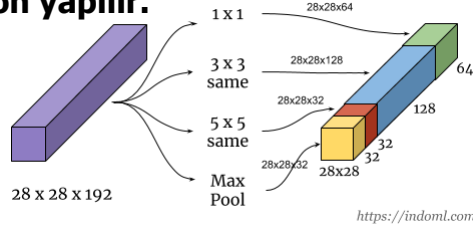


42

CNN mimarileri

GoogLeNet

- ▶ Her katmanda birden fazla convolution ve max pooling yapılır.
- ▶ Her katmanın sonunda **convolution** ve **max pooling** sonuçları **concatenation** yapılır.

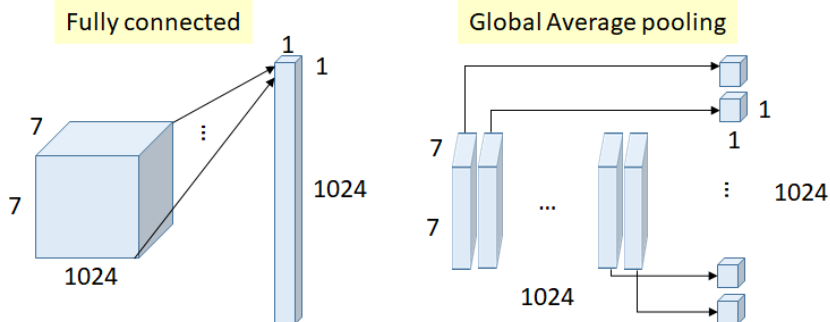


43

CNN mimarileri

GoogLeNet

- ▶ Diğer CNN uygulamalarından farklı olarak **FC katmanı** sadece en son katmanda değil, **ara katmanlarda da yer alır.**

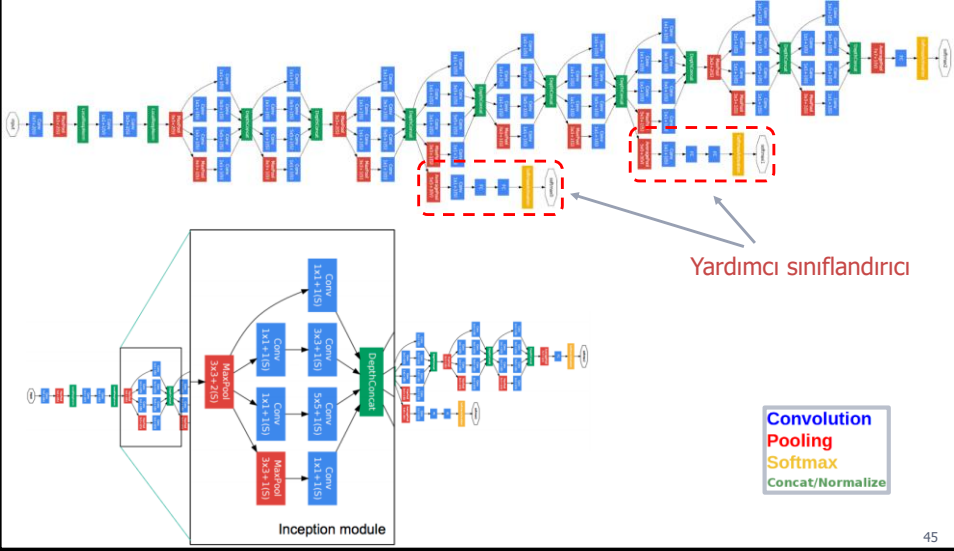


44

CNN mimarileri

GoogLeNet

- ▶ GoogLeNet katman yapısı aşağıdaki gibidir.



45

İçerik

- ▶ Training
- ▶ Değerlendirme
- ▶ CNN mimarileri
- ▶ CNN uygulamaları

46

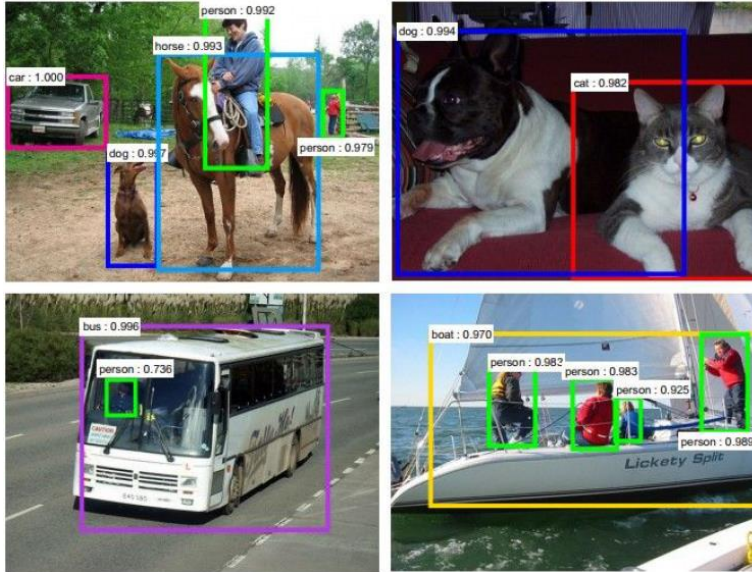
CNN uygulamaları

- ▶ CNN, **görüntüyle ilgili problemlerde başarıyla uygulanan** bir modeldir.
- ▶ CNN, **tavsiye sistemleri, NLP** ve birçok **farklı alanda başarı ile uygulanmıştır.**
- ▶ CNN, giriş verisindeki **önemli özellikleri kendisi otomatik olarak algılar.**
- ▶ CNN modeli, **insandan daha iyi ve daha hızlı** görüntü sınıflandırabilmektedir.
- ▶ CNN modeli çok hızlı ve **yüksek doğrulukta nesne tanımlayabilmektedir.**

47

CNN uygulamaları

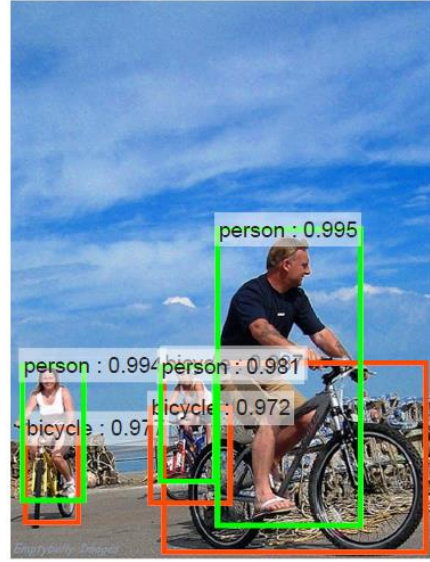
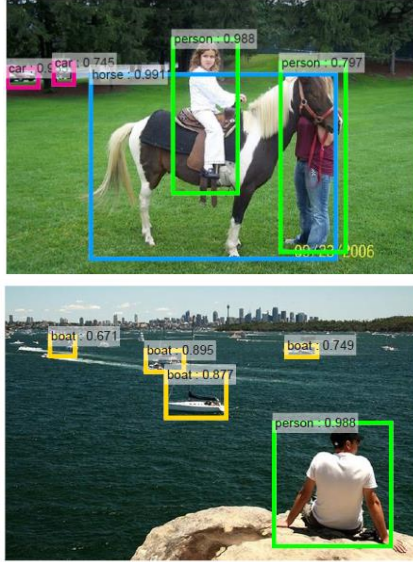
- ▶ Nesne tanımlama



48

CNN uygulamaları

► Nesne tanımlama

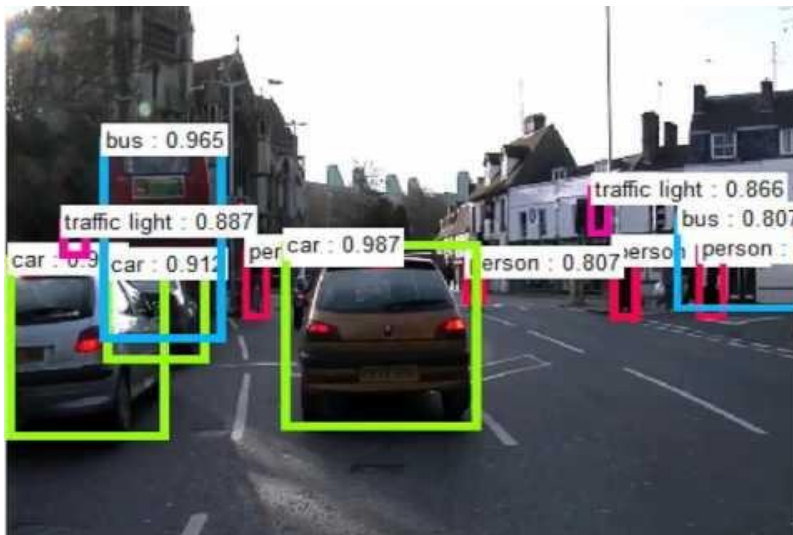


49

CNN uygulamaları

► Gerçek zamanlı nesne tanımlama (CNN)

<https://www.youtube.com/watch?v=WZmSMkK9VuA>



50

CNN uygulamaları

- Gerçek zamanlı nesne tanımlama (CNN)
<https://youtu.be/70Kv8Rr72ag>



51

CNN uygulamaları

- Görüntü renklendirme (CNN)
<https://youtu.be/ys5nMO4Q0iY>



52

CNN uygulamaları

- Sürücüsüz otomobil (CNN+RNN)
<https://youtu.be/hLaEV72elj0>



53

CNN uygulamaları

- Robotik (CNN+RNN+FLC) - <https://youtu.be/tf7IEVTDjng>



54

CNN uygulamaları

- ▶ Robotik (CNN+RNN+FLC)

<https://www.youtube.com/watch?v=kgaO45SyaO4>



55

Ödev

- ▶ Multimodal convolutional neural network ile kamera görüntülerindeki nesnelere için konum belirlemeye yönelik detaylı bir araştırma ödevi hazırlayınız.

56