BİL 475 Örüntü Tanıma

Hafta-9:

Yapay Sinir Ağları-2

Yapay Sinir Ağları

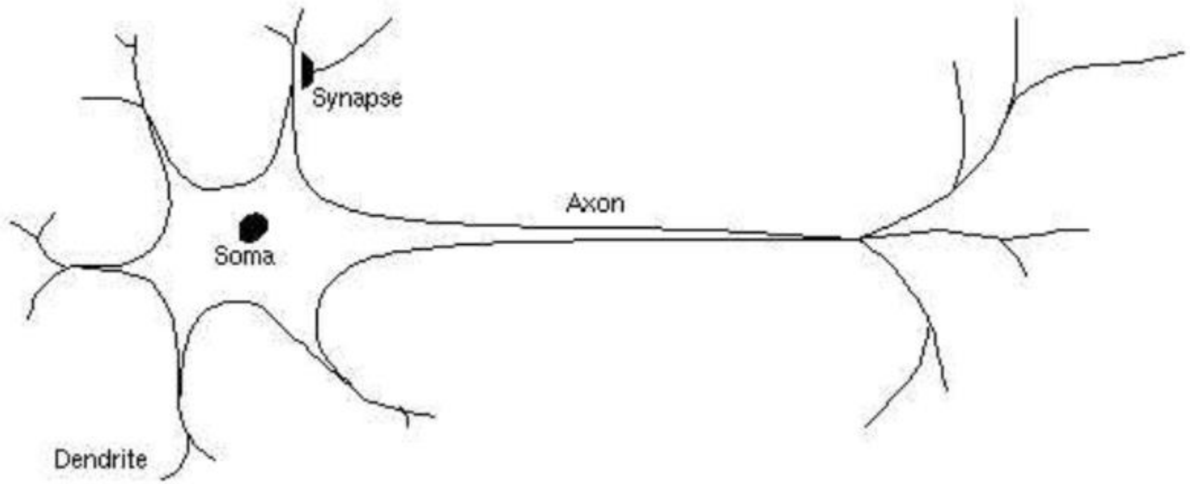
Bulletin of Mathematical Biology Vol. 52, No. 1/2, pp. 99–115, 1990.
Printed in Great Britain.

0092–8240/90$3.00 + 0.00
Pergamon Press plc
Society for Mathematical Biology

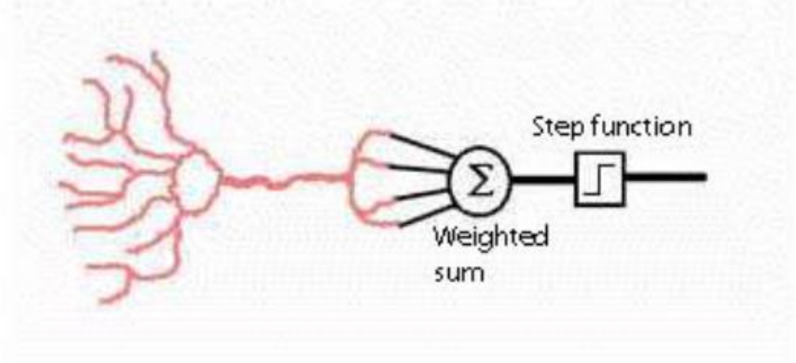
A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY*

■ WARREN S. MCCULLOCH AND WALTER PITTS
University of Illinois, College of Medicine,
Department of Psychiatry at the Illinois Neuropsychiatric Institute,
University of Chicago, Chicago, U.S.A.

Because of the "all-or-none" character of nervous activity, neural events and the relations among them can be treated by means of propositional logic. It is found that the behavior of every net can be described in these terms, with the addition of more complicated logical means for nets containing circles; and that for any logical expression satisfying certain conditions, one can find a net behaving in the fashion it describes. It is shown that many particular choices among possible neurophysiological assumptions are equivalent, in the sense that for every net behaving under one assumption, there exists another net which behaves under the other and gives the same results, although perhaps not in the same time. Various applications of the calculus are discussed.



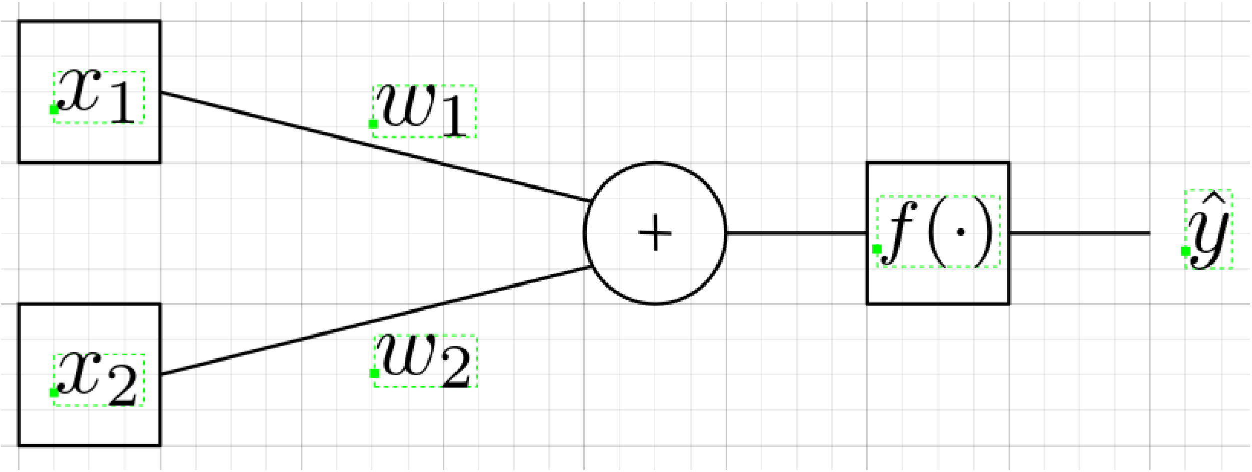
(Fig. 1) A biological neuron

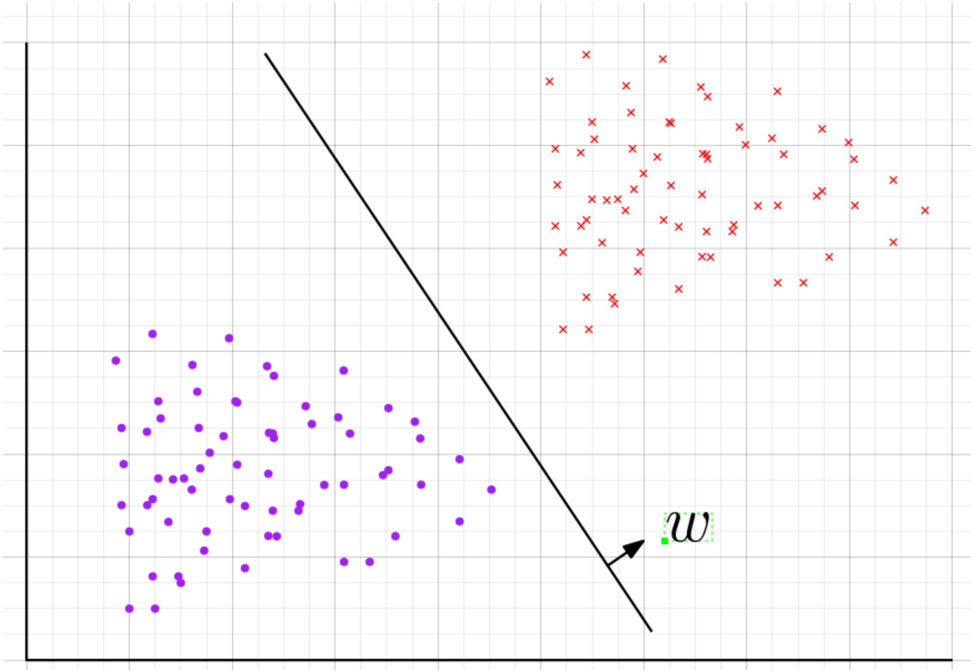


(Fig. 2) An artificial neuron (perceptron)

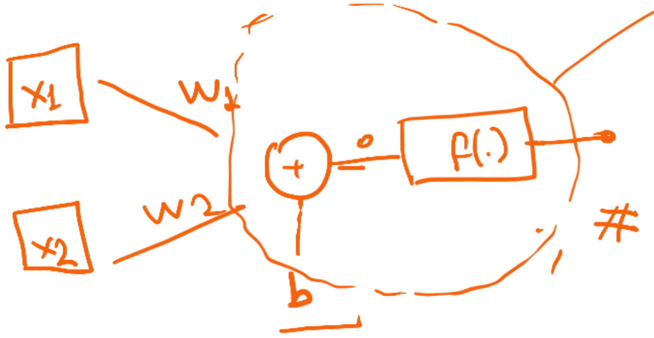
https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/Neuron/index.html

Yapay Sinir Ağları





Yapay Sinir Ağları

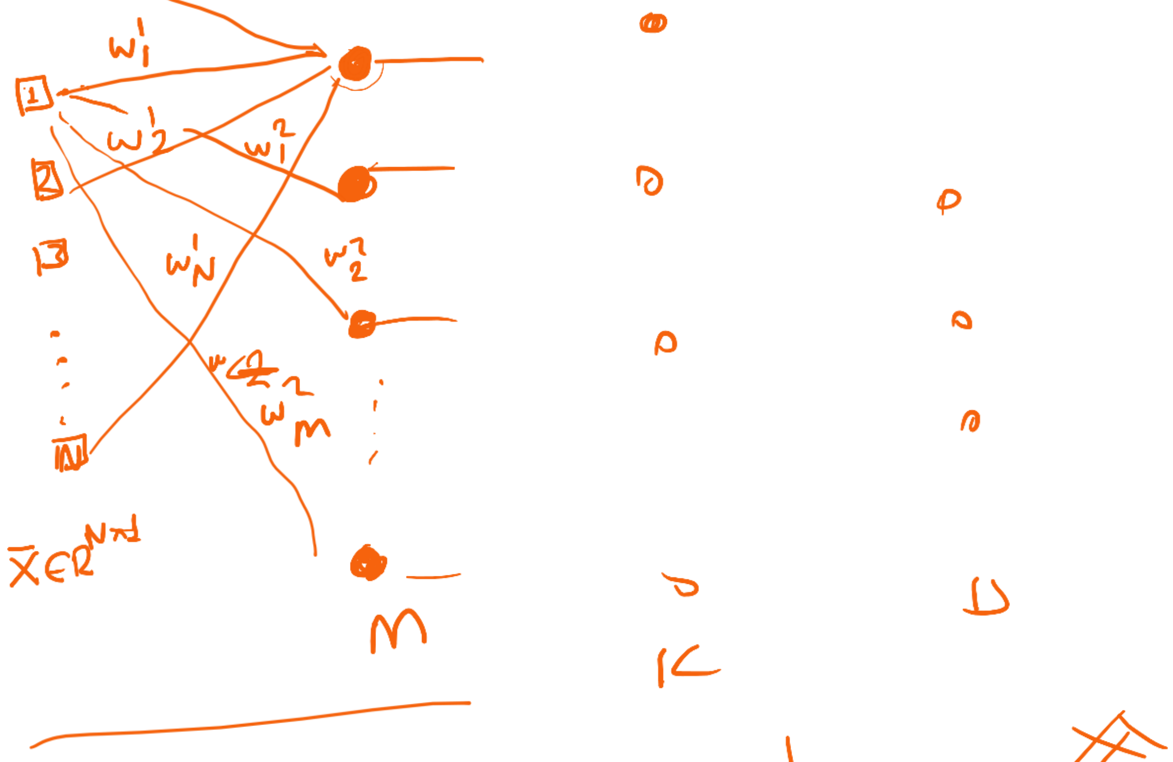


Perceptron

$$f(w_1 x_1 + w_2 x_2) = p_1 \quad \hat{y}$$

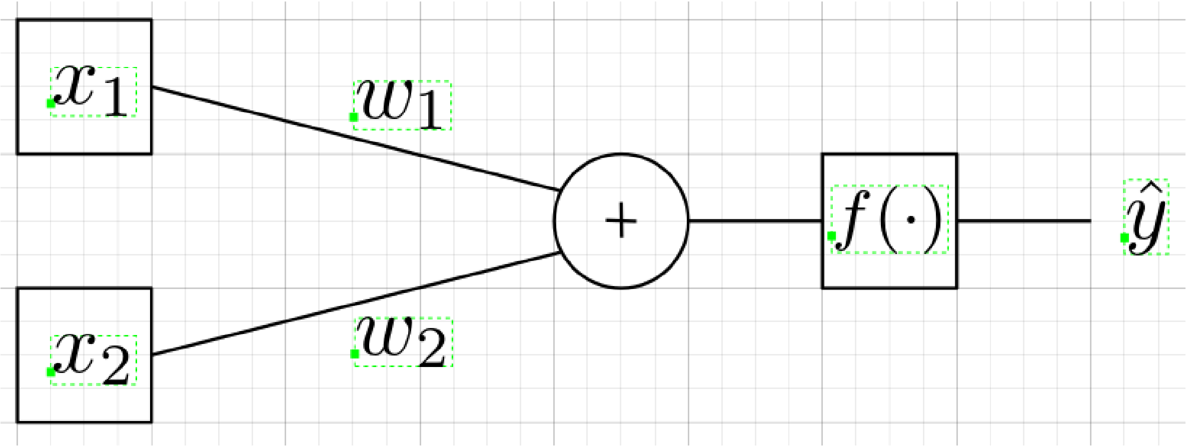
$$f(\langle \bar{w}, \bar{x} \rangle) = \hat{y}, p$$

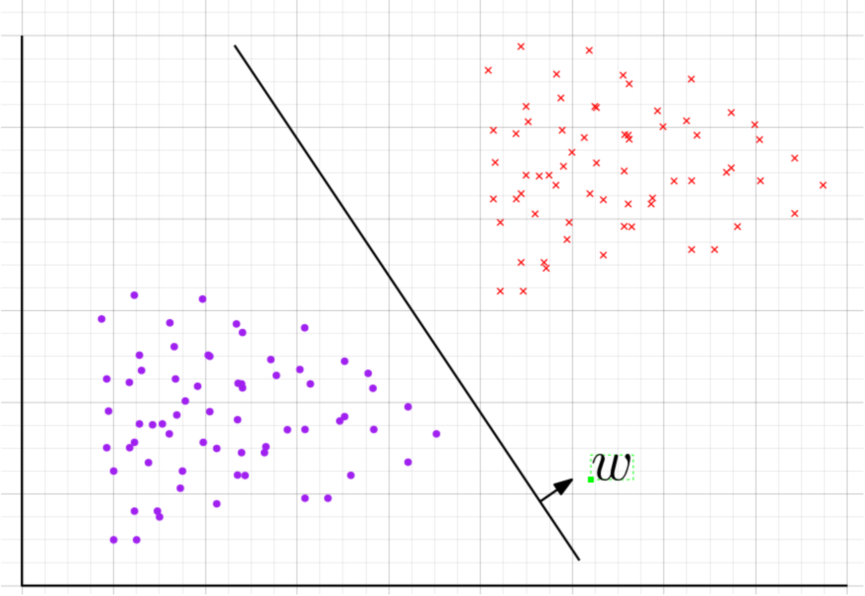
$$f(\langle \bar{w}, \bar{x} \rangle + b) = \hat{y}$$



Multilayer Perceptron

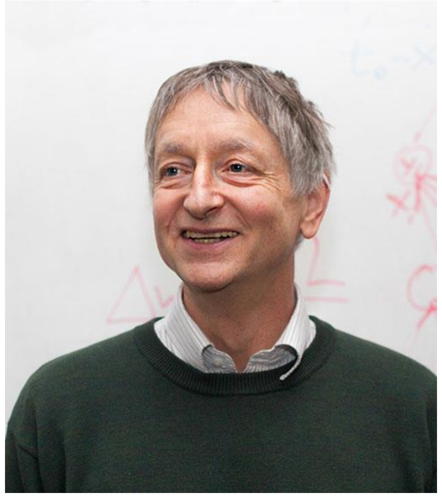
Yapay Sinir Ağları





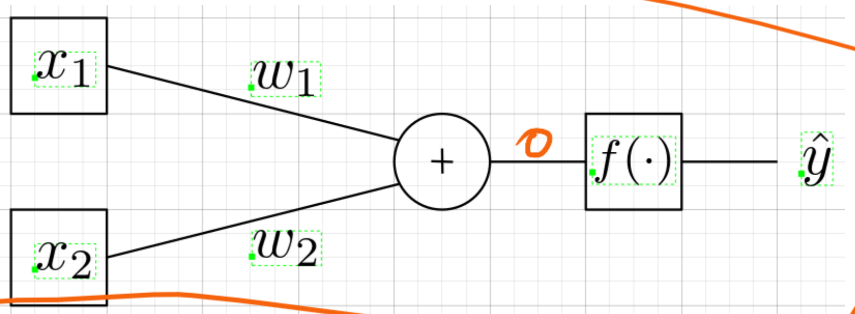
Hatanın Geriye Yayılımı

Geoffrey Hinton



$\frac{1}{1+e^{-x}}$

Yapay Sinir Ağları: Hatanın Geriye Yayılması



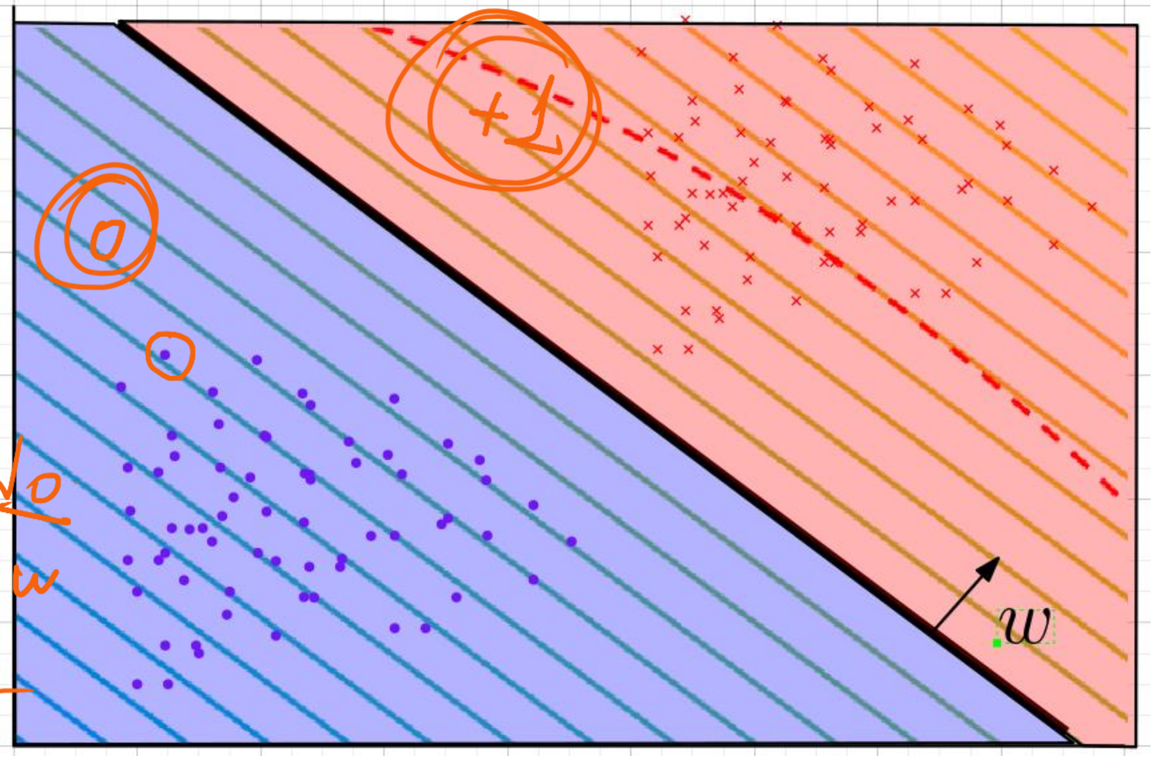
$$\hat{y}_i = f(\langle \boldsymbol{w}, \boldsymbol{x_i} \rangle)$$

o

$$e_i = \frac{1}{2}(\hat{y}_i - y_i)^2$$

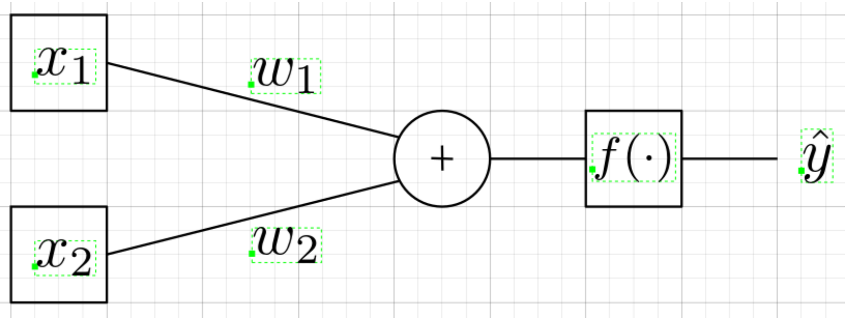
$$\frac{de_i}{d\boldsymbol{w}} = (\hat{y}_i - y_i)\frac{d\hat{y}_i}{d\boldsymbol{w}}$$

$$\frac{de_i}{d\bar{w}} = \frac{de_i}{d\hat{y}_i} \cdot \frac{d\hat{y}_i}{do} \cdot \frac{do}{dw}$$



$$w_{t+1} = w_t - \mu \frac{\partial e_i}{\partial \bar{w}_t}$$

Yapay Sinir Ağları: Hatanın Geriye Yayılması



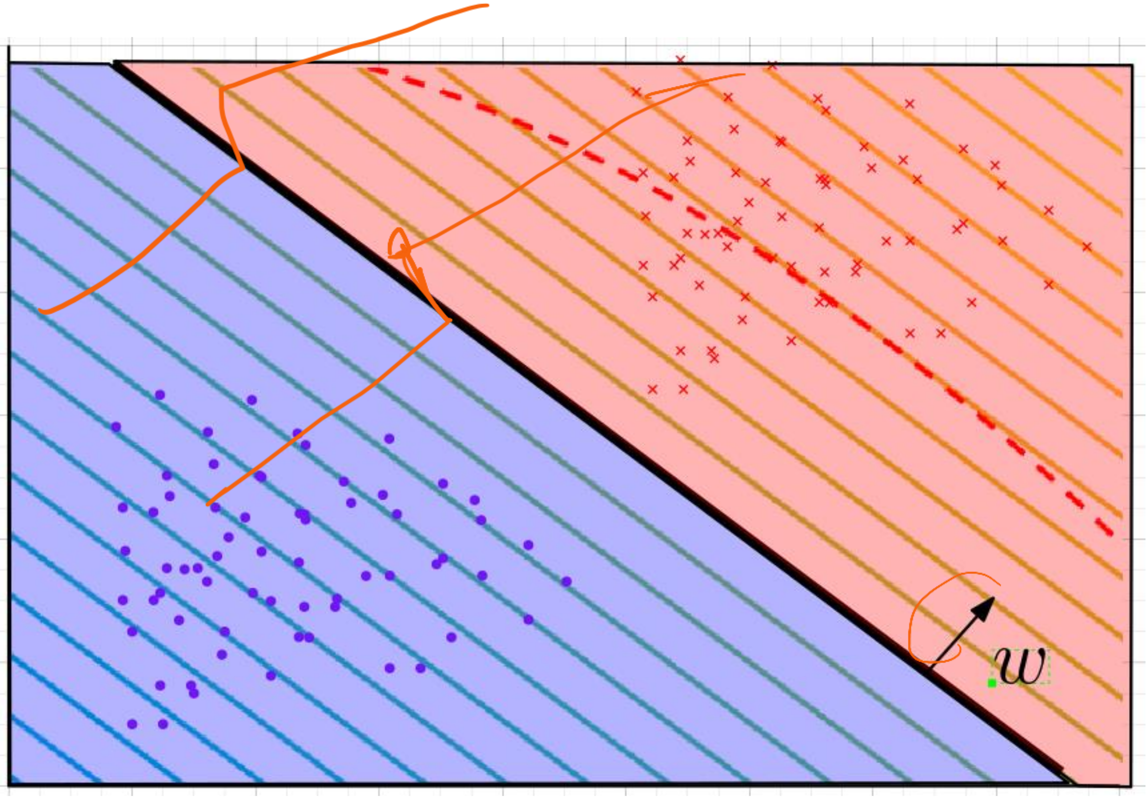
$$\hat{y}_i = f(\langle \boldsymbol{w}, \boldsymbol{x_i} \rangle)$$

$$\frac{d\hat{y}_i}{d\bar{w}} = \frac{df(o)}{do} \cdot \frac{do}{d\bar{w}}$$

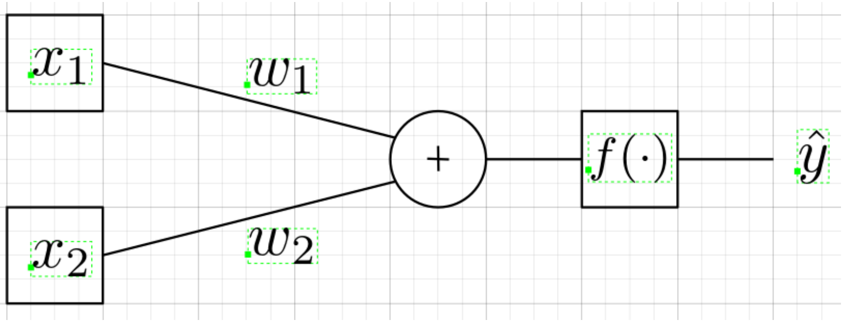
$$e_i = \frac{1}{2}(\hat{y}_i - y_i)^2$$

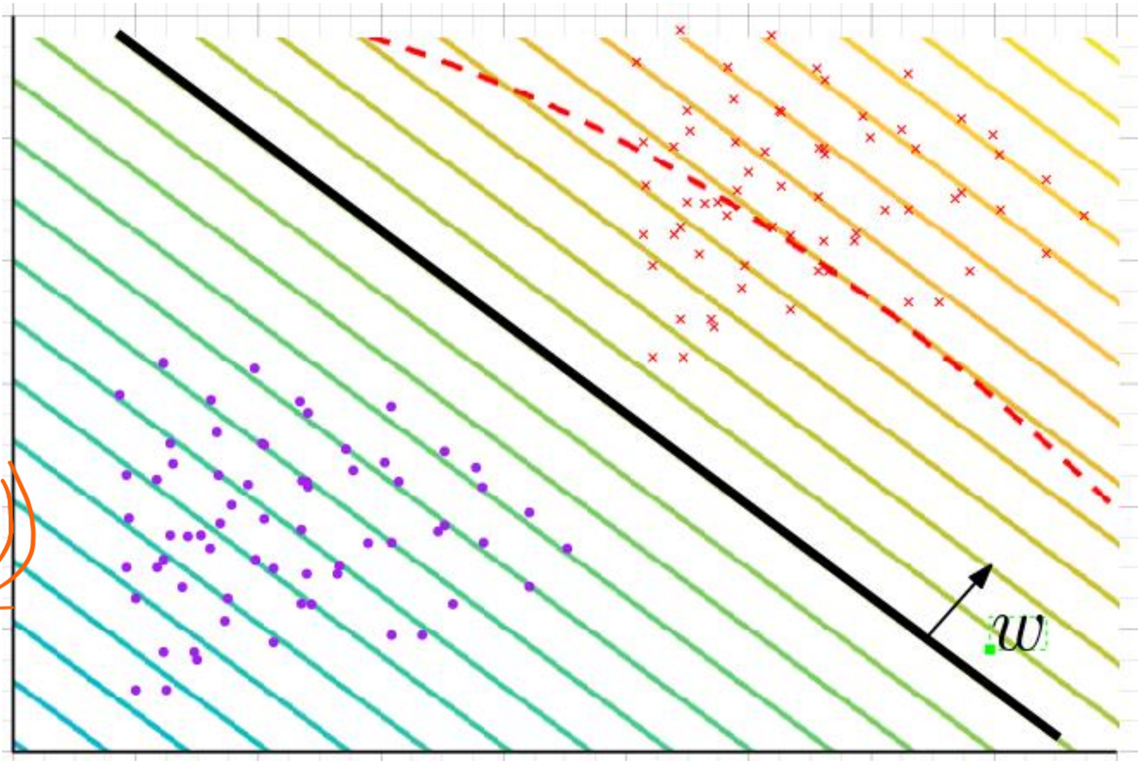
$$\frac{de_i}{d\boldsymbol{w}} = (\hat{y}_i - y_i)\frac{d\hat{y}_i}{d\boldsymbol{w}}$$

$$\frac{d\, f(x)}{dx}$$



Yapay Sinir Ağları: Hatanın Geriye Yayılması





$$\hat{y}_i = f(\langle \boldsymbol{w}, \boldsymbol{x_i} \rangle)$$

$$\frac{d}{do} f(o) = ? \quad f(o).(1-f(o))$$

$$e_i = \frac{1}{2}(\hat{y}_i - y_i)^2$$

$$\frac{de_i}{d\boldsymbol{w}} = (\hat{y}_i - y_i)\frac{d\hat{y}_i}{d\boldsymbol{w}}$$

$$o = \langle \bar{w}, \bar{x}_i \rangle$$

$$\frac{d\,f(x)}{dx}$$

$$f(x) = \frac{1}{1+e^{-x}} \quad // = f(x).(1-f(x))$$

$$\frac{do}{d\bar{w}} = \bar{x}_i$$

$$\frac{de_i}{d\bar{w}} = \underbrace{\frac{de_i}{d\hat{y}_i}}_{(\hat{y}_i - y_i)} \cdot \underbrace{\frac{d\hat{y}_i}{do}}_{(\hat{y}_i \cdot (1-\hat{y}_i))} \cdot \underbrace{\frac{do}{d\bar{w}}}_{\bar{x}_i}$$

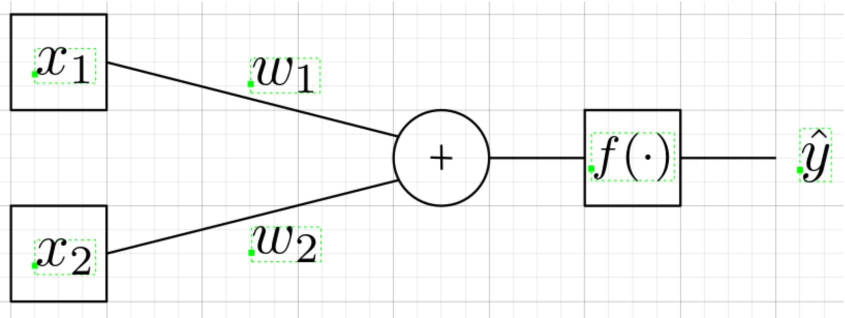
$\hat{y}_i = f(o)$

$o = \langle \bar{w}, \bar{x}_i \rangle$

$\alpha \cdot \bar{x}_i$

Katsayı

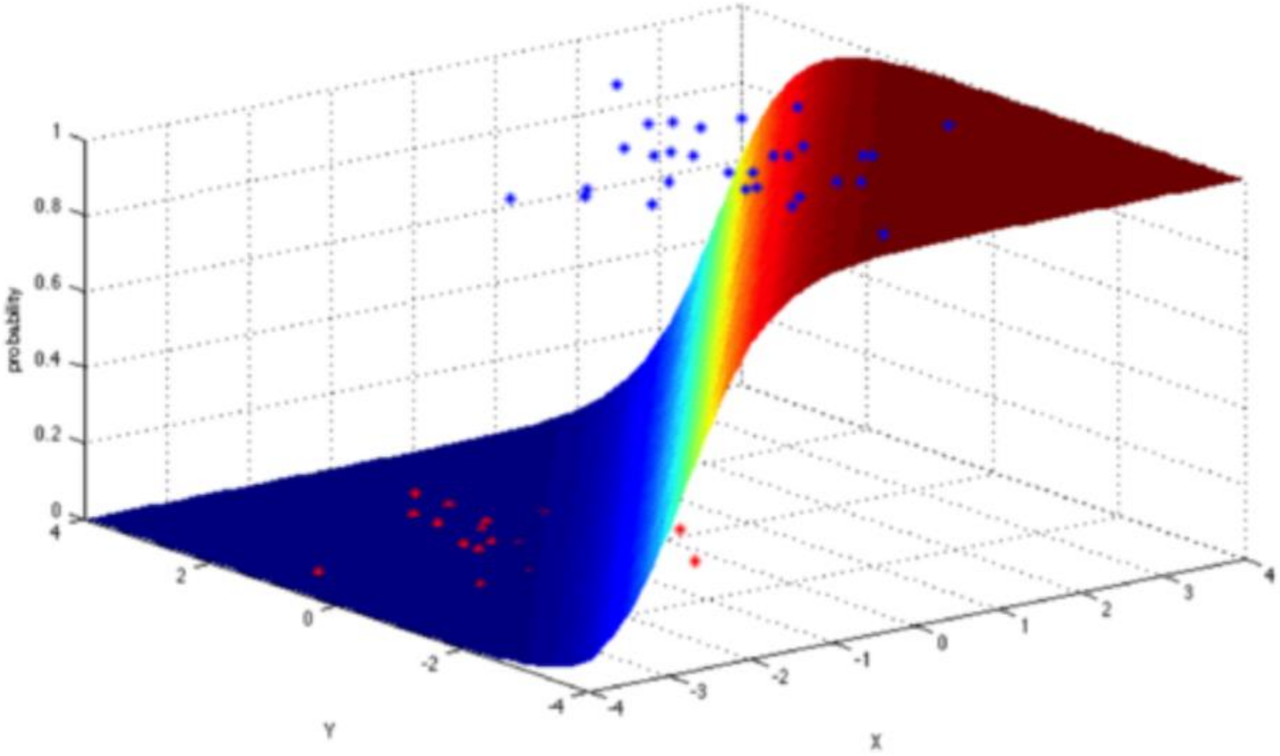
vektör

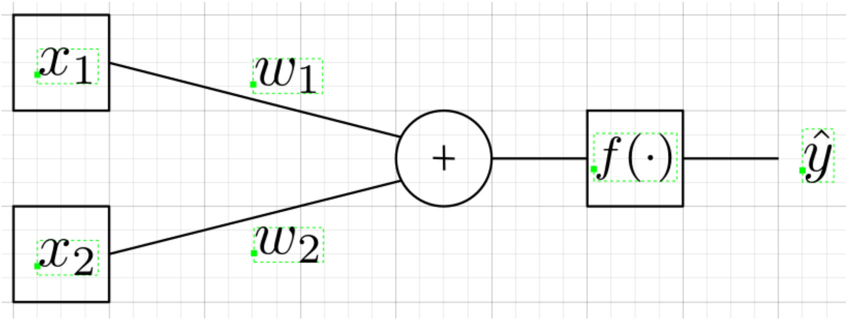


$$\mathcal{L}\{\ f(\boldsymbol{x}_i;\boldsymbol{w})\ \ ,y_i\ \}\qquad\qquad y_i \in \{0,1\}$$

Lojistik Regresyon

$$P\{y=1|\boldsymbol{x}\} = \frac{1}{1+e^{-\langle \boldsymbol{w},\boldsymbol{x}\rangle}}$$





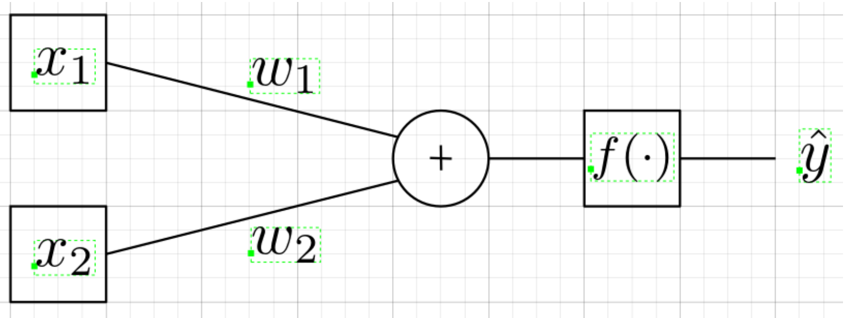
Lojistik Regresyon

$$P\{y = 1|\boldsymbol{x}\} = \frac{1}{1 + e^{-\langle \boldsymbol{w},\boldsymbol{x}\rangle}}$$

$$\mathcal{L}\{\ f(\boldsymbol{x}_i; \boldsymbol{w})\ \ , y_i\ \}\qquad y_i \in \{0,1\}$$

$$o_i = \langle \boldsymbol{w}, \boldsymbol{x_i} \rangle \qquad \hat{y}_i = \sigma(\ o_i\ ) \qquad e_i = \frac{1}{2}(\hat{y}_i - y_i)^2$$

$$\frac{de_i}{d\boldsymbol{w}} = \frac{de_i}{d\hat{y}_i}\ \frac{d\hat{y}_i}{do_i}\ \frac{do_i}{d\boldsymbol{w}}$$



Lojistik Regresyon

$$P\{y=1|\boldsymbol{x}\} = \frac{1}{1+e^{-\langle \boldsymbol{w},\boldsymbol{x}\rangle}}$$

$$\mathcal{L}\{\ f(\boldsymbol{x}_i;\boldsymbol{w})\ \ ,y_i\ \} \qquad y_i \in \{0,1\}$$

$$o_i = \langle \boldsymbol{w}, \boldsymbol{x_i}\rangle \qquad \hat{y}_i = \sigma(\,o_i\,) \qquad e_i = \frac{1}{2}(\hat{y}_i - y_i)^2$$

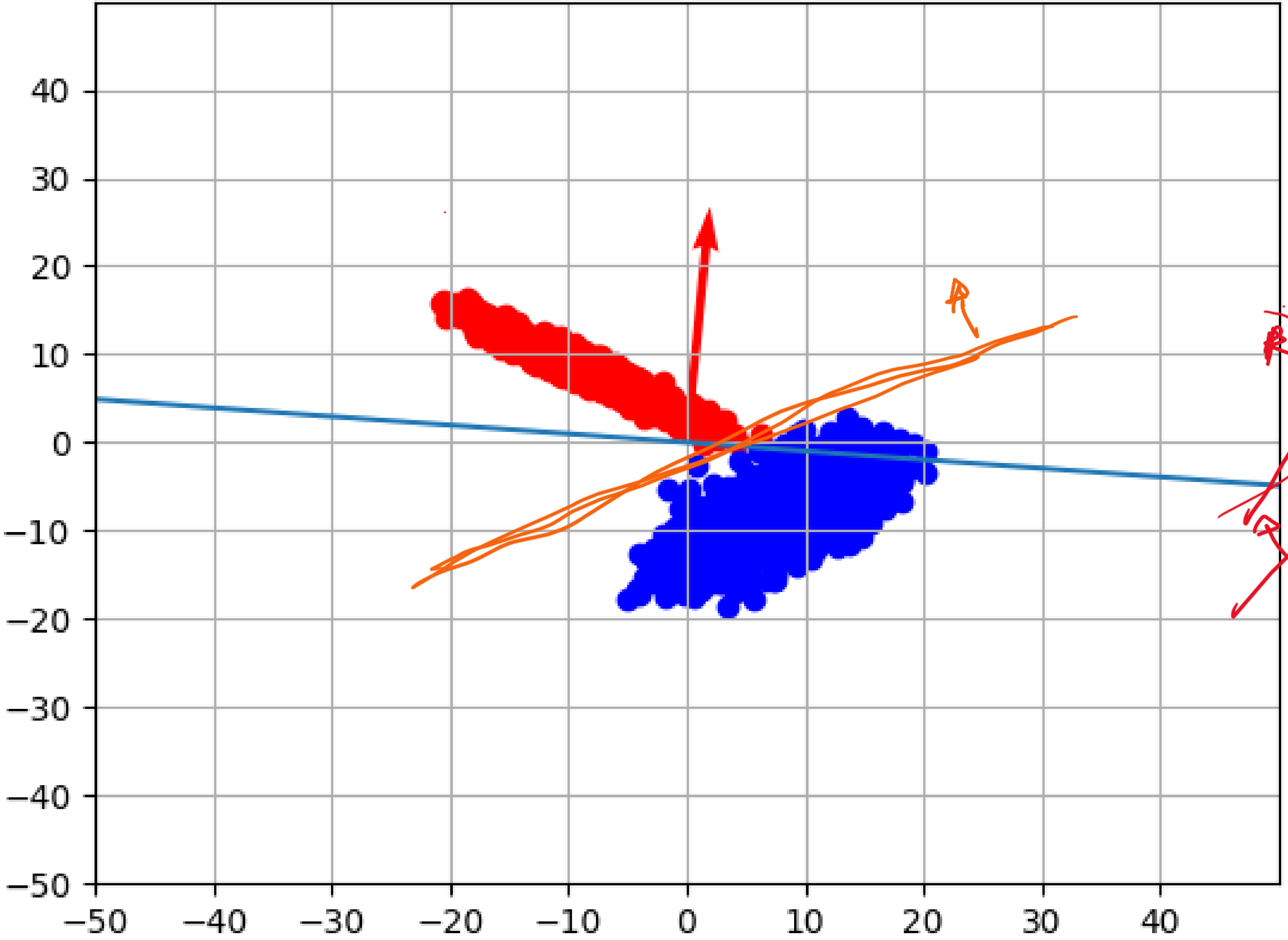
$$\frac{de_i}{d\boldsymbol{w}} = \frac{de_i}{d\hat{y}_i}\,\frac{d\hat{y}_i}{do_i}\,\frac{do_i}{d\boldsymbol{w}}$$

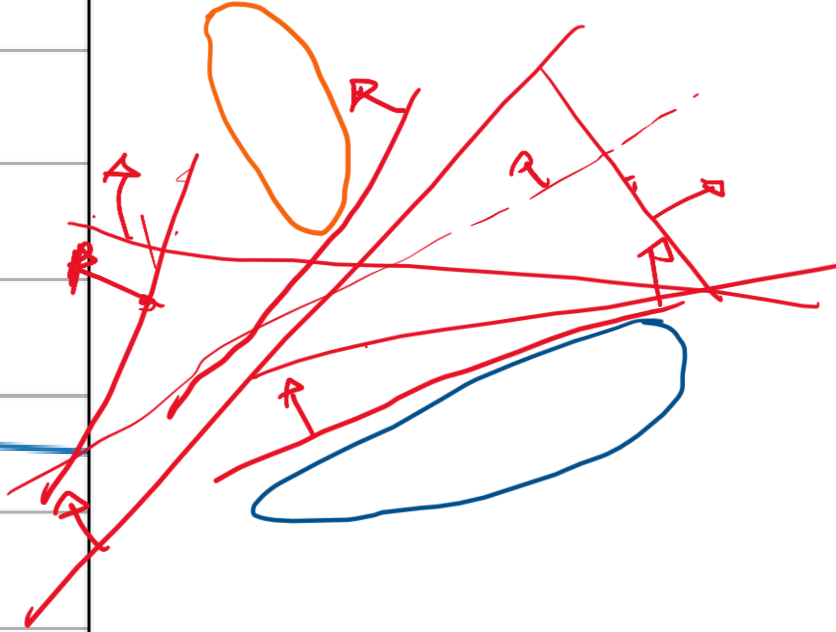
$$E = \frac{1}{N}\sum_{i=1}^{N} e_i$$

$$\frac{de_i}{d\boldsymbol{w}} = (\hat{y}_i - y_i)\,\hat{y}_i(1-\hat{y}_i)\,\boldsymbol{x_i}$$

$$w_{t+1} = w_t - \mu \cdot \frac{de_i}{dw}$$

yılması





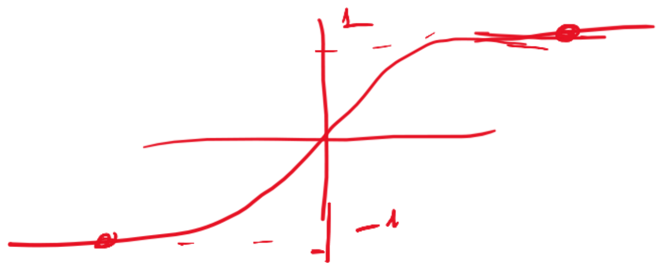
Gradyanların Sönümlenmesi

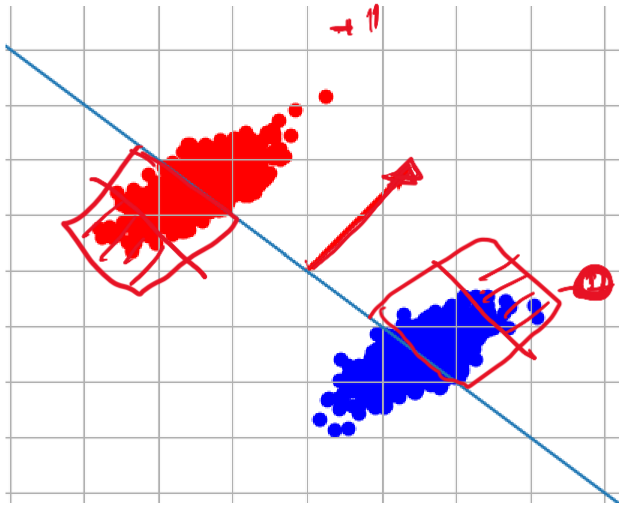
0 $\quad 1-0$

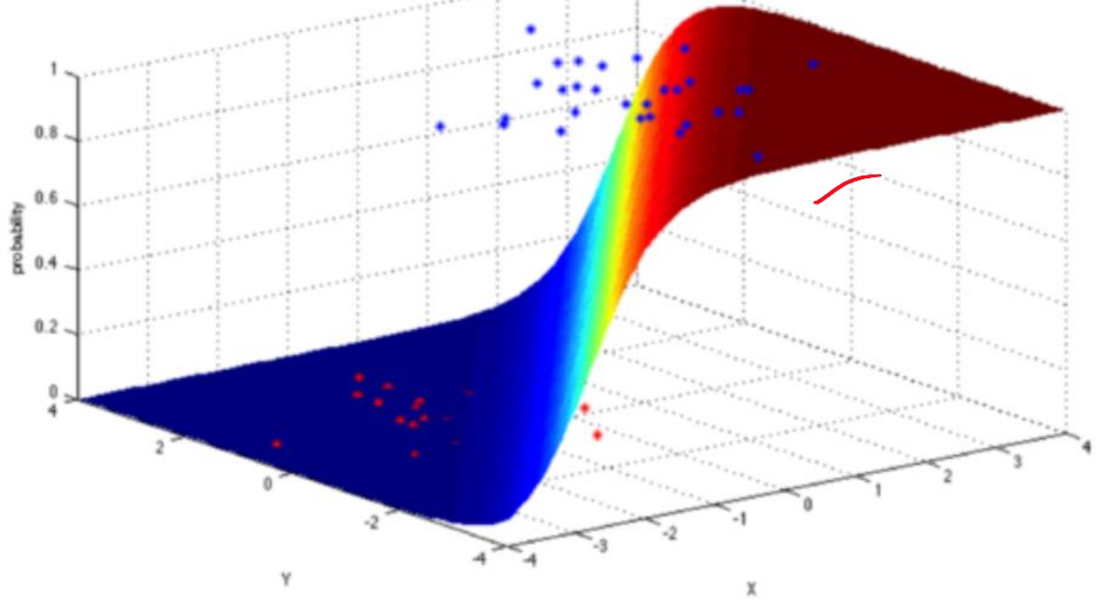
$$\frac{de_i}{d\mathbf{w}} = (\hat{y}_i - y_i)\hat{\mathbf{y}}_i(\mathbf{1} - \hat{\mathbf{y}}_i)\,\mathbf{x}_i$$

0 $\quad$ 1

0







https://towardsdatascience.com/nothing-but-numpy-understanding-creating-binary-classification-neural-networks-with-e746423c8d5c

$\bar{x}_i \rightarrow \square \rightarrow \hat{y}_i$, $P_i = 0$ $\qquad \bar{x}_i \qquad +1$

$\hat{y}_i = 0$ $\qquad$ 0

$\hat{y}_i = 1$

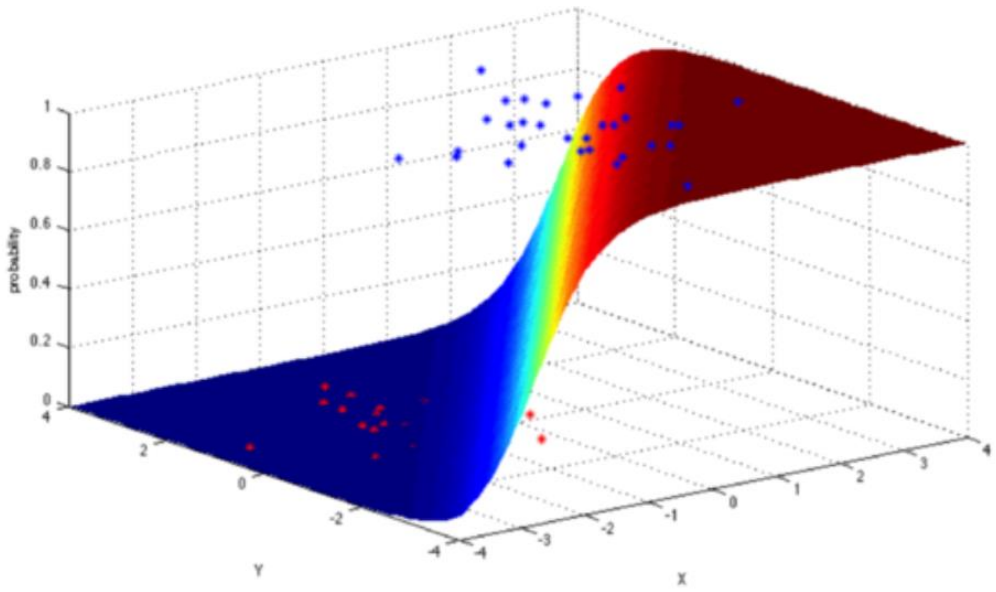
$\frac{1}{2}(0-1)^2$

$(1-0)\boxed{1.(1-1)}\,\bar{x}_i$

Gradyanların Sönümlenmesi

Cross Entropy Loss

$$\frac{de_i}{d\boldsymbol{w}} = (p_i - y_i)\, \boldsymbol{p_i(1 - p_i)}\, \boldsymbol{x_i}$$



$$\mathcal{L}_i(p_i, y_i) = -(y_i \log(p_i) + (1 - y_i) \log(1 - p_i))$$

$$\frac{d\mathcal{L}_i(p_i, y_i)}{dp_i} = -\left(\frac{y_i}{p_i} - \frac{1 - y_i}{1 - p_i}\right) = \left(-\frac{y_i}{p_i} + \frac{1 - y_i}{1 - p_i}\right)$$

$$\frac{de_i}{d\boldsymbol{w}} = \boldsymbol{(p_i - y_i)\, p_i(1 - p_i)\, x_i}$$

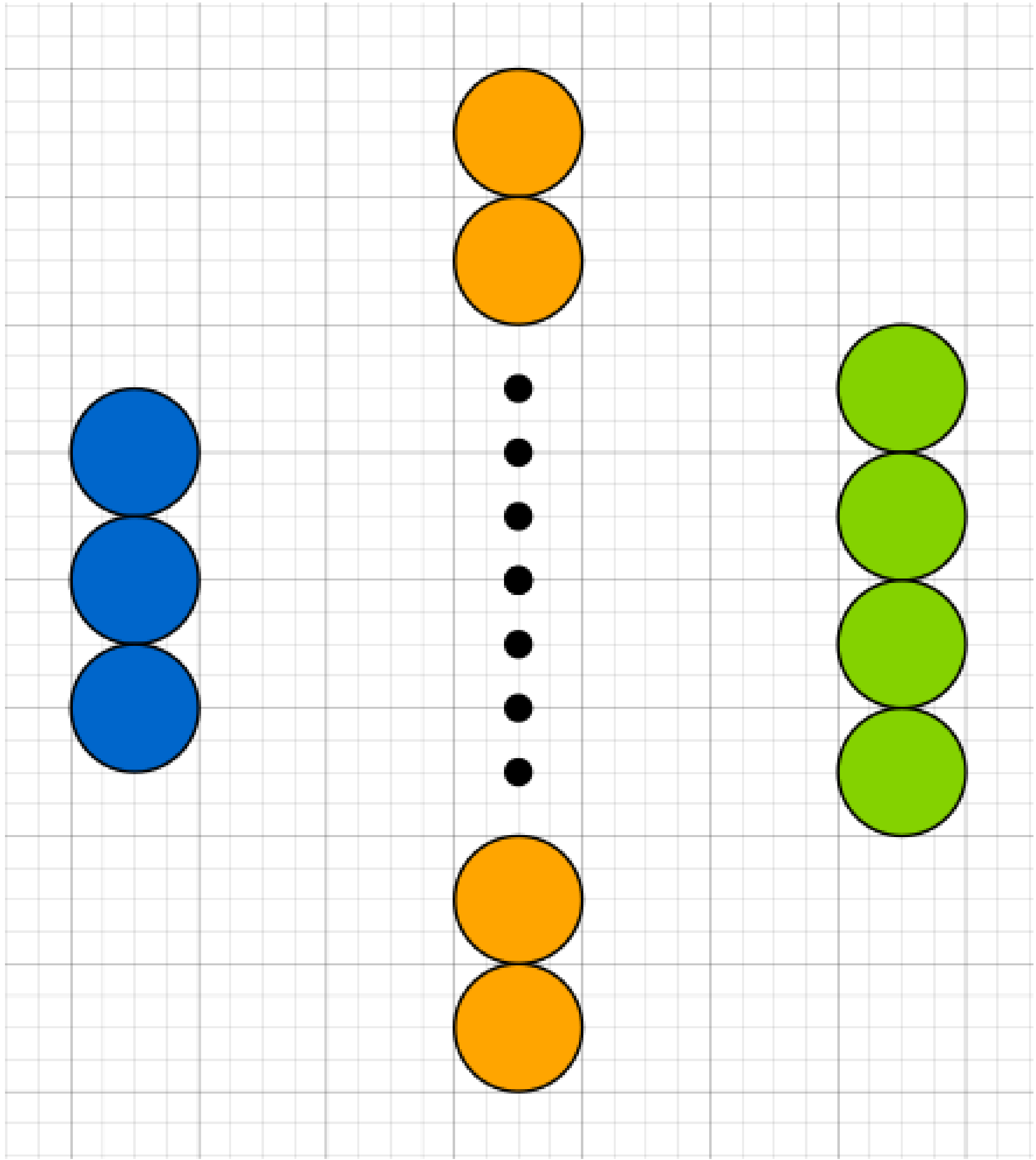
$$\frac{de_i}{do_i} = \begin{cases} -(1 - p_i) & y_i = 1 \\ p_i & y_i = 0 \end{cases}$$

1 Nöron YSA Özeti

- N boyutlu uzayda bir üst düzlem
- Aktivasyon fonksiyonu doğrusal olmamalıdır
- GD ile eğitim yapılır
- Sınıflandırma için Cross Entropy Loss kullanılır.

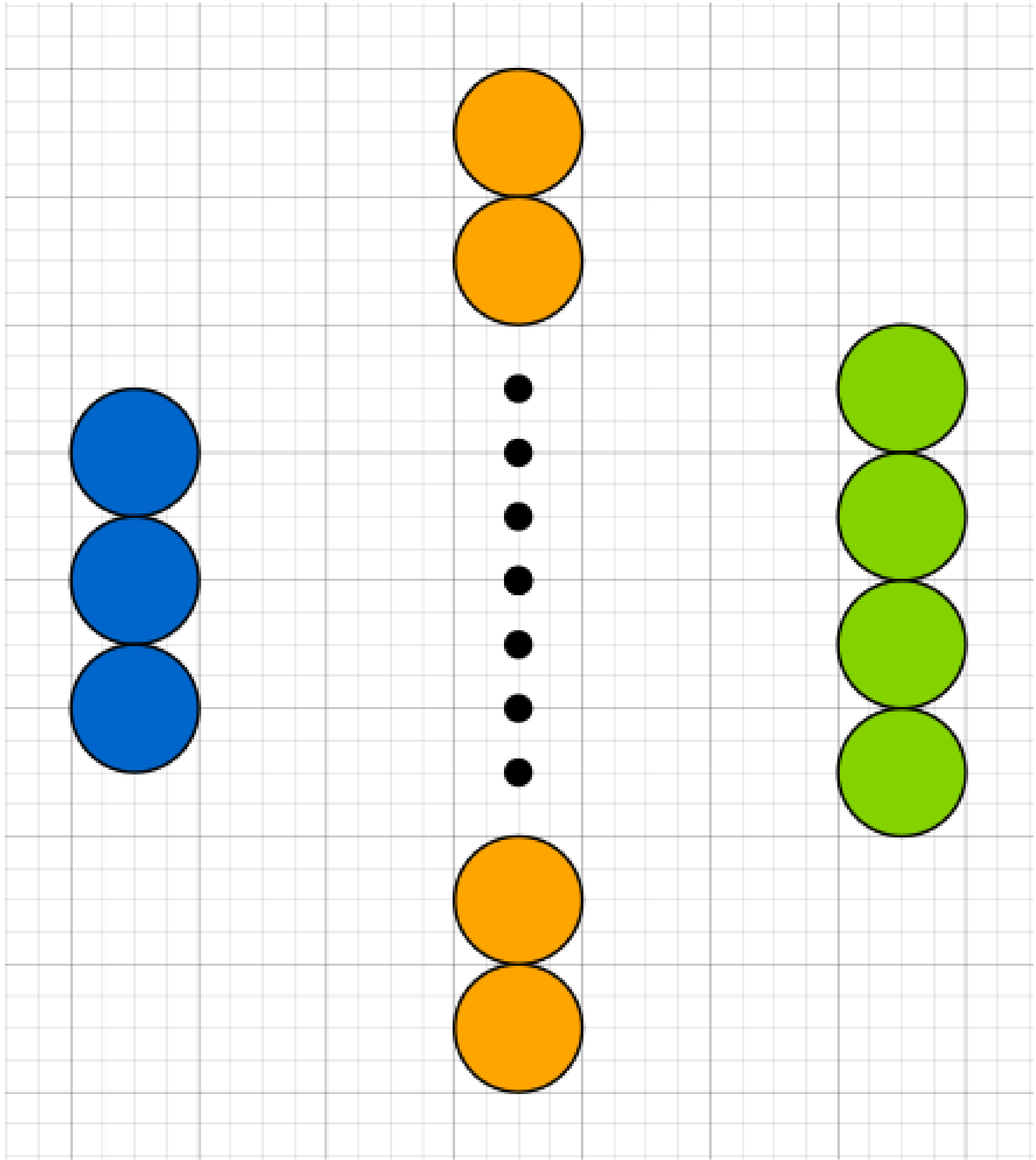
Multi Layer Perceptron (MLP)

Multi Layer Perceptron (MLP)



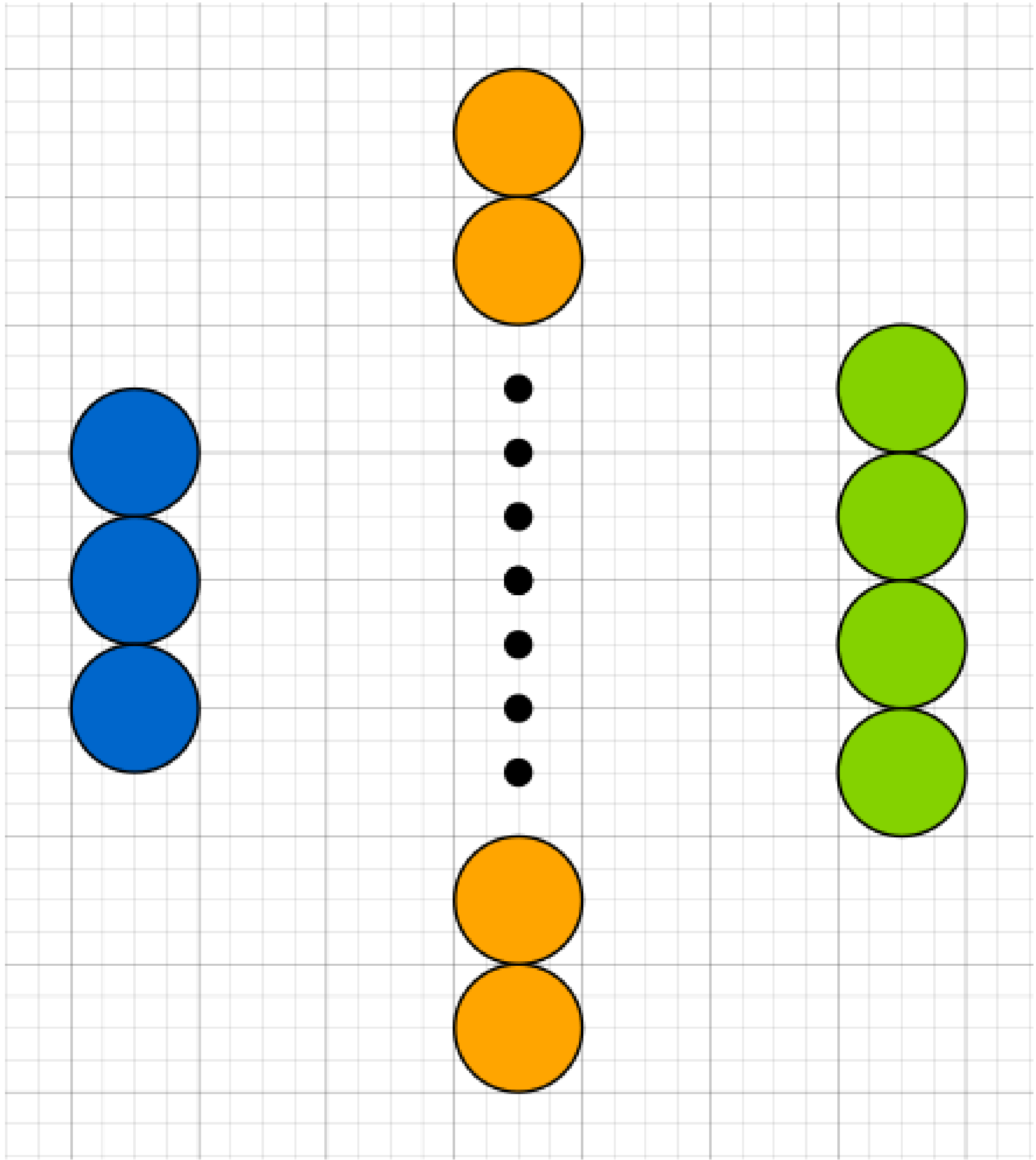
http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

Multi Layer Perceptron (MLP)



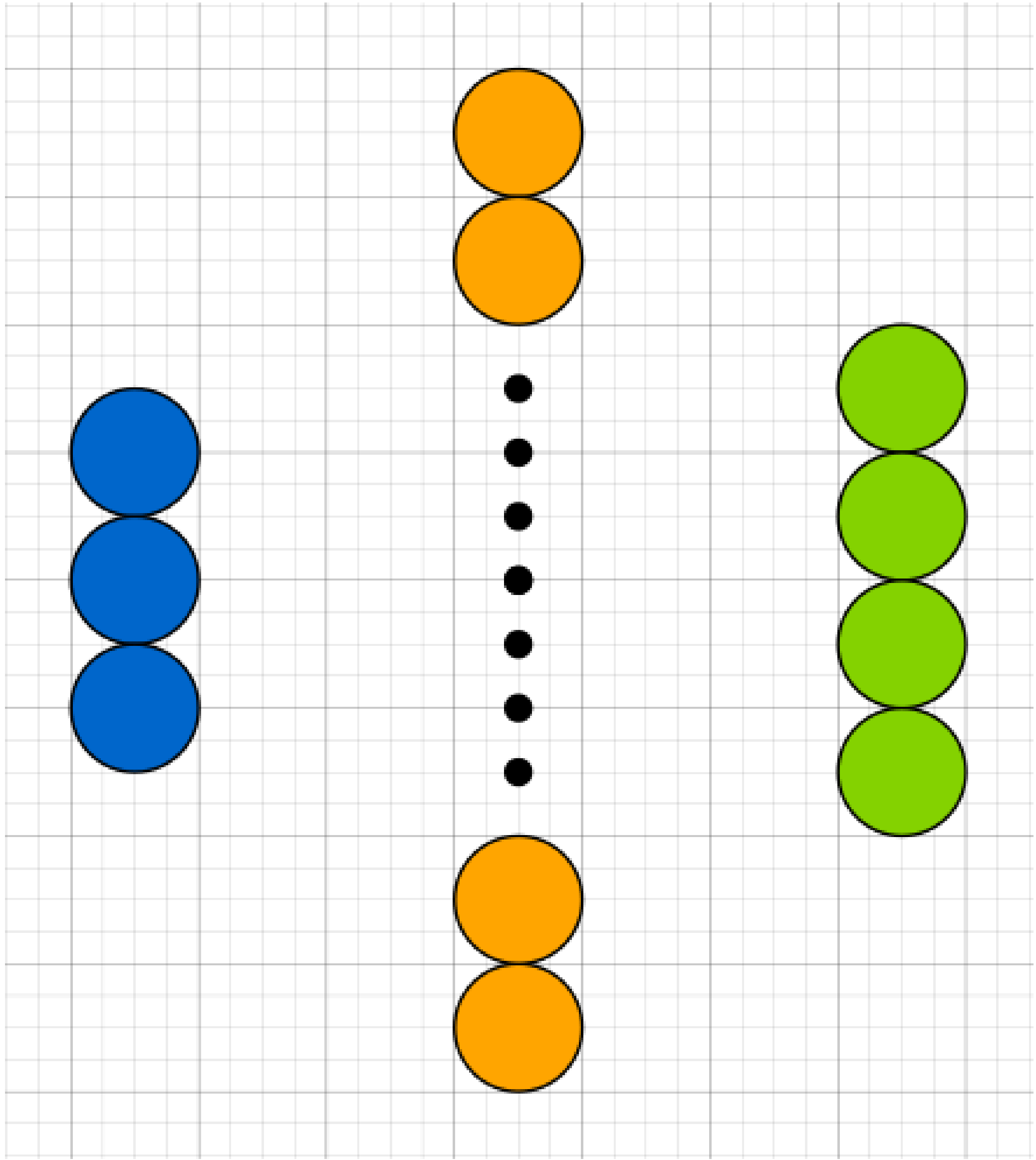
Tahmin – Hedef İlişkisi (ℓ_2)

Multi Layer Perceptron (MLP)



Tahmin – Hedef İlişkisi (Çapraz Entropi)

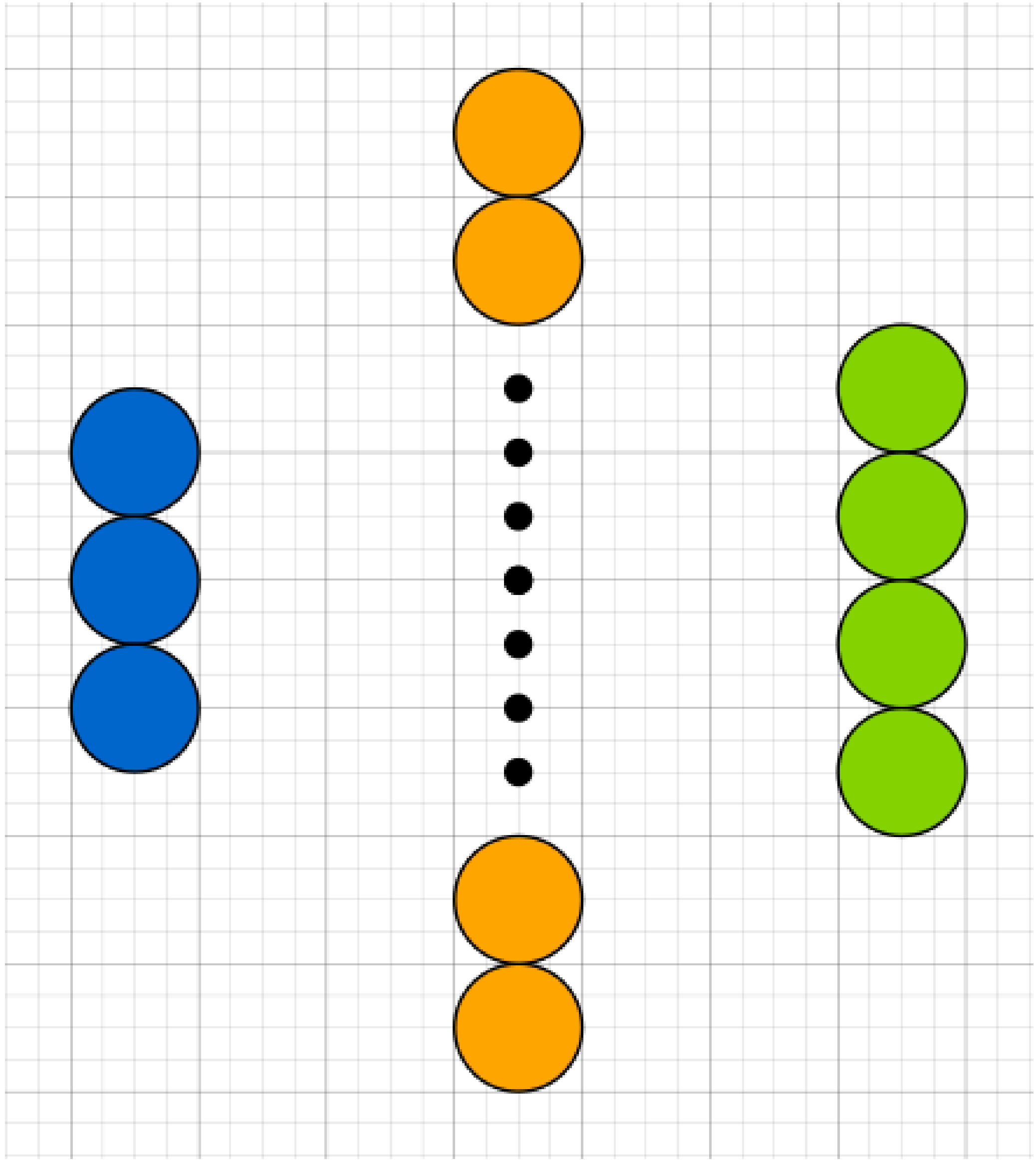
Multi Layer Perceptron (MLP)



Tahmin – Hedef İlişkisi (Çapraz Entropi)

Kullback-Leibler Divergance

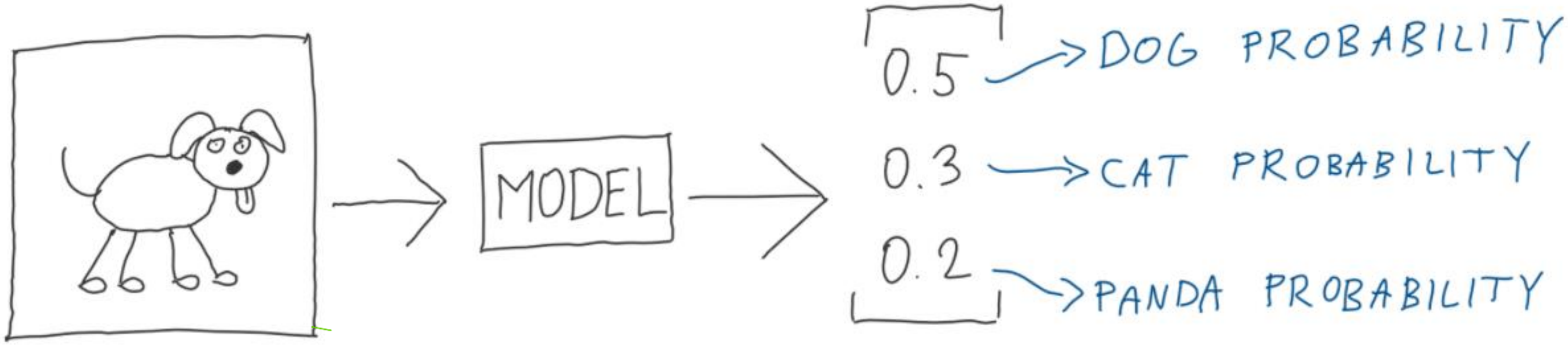
Multi Layer Perceptron (MLP)



Tahmin – Hedef İlişkisi (Çapraz Entropi)

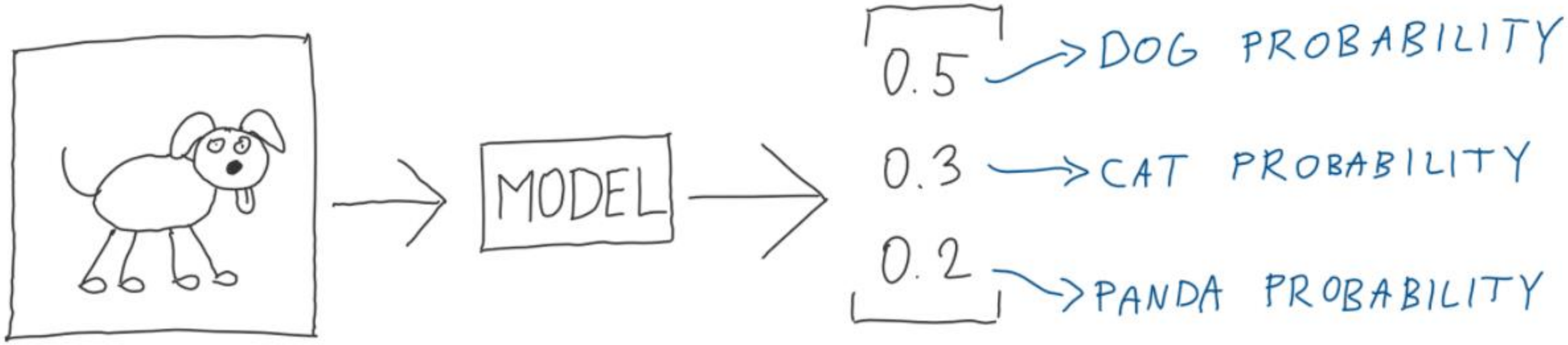
Kullback-Leibler Divergance

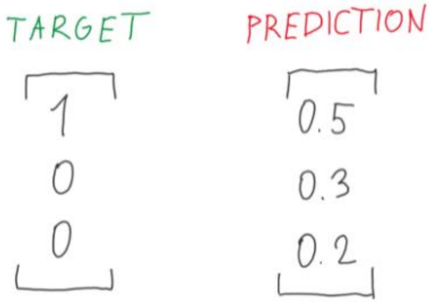
Multi Layer Perceptron (MLP)



https://towardsdatascience.com/cross-entropy-for-classification-d98e7f974451

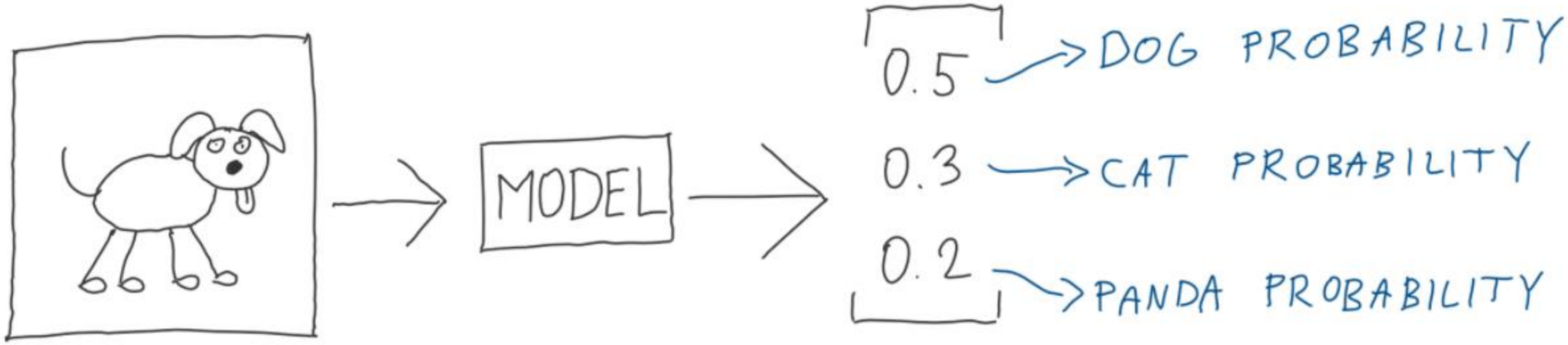
Multi Layer Perceptron (MLP)





https://towardsdatascience.com/cross-entropy-for-classification-d98e7f974451

Multi Layer Perceptron (MLP)

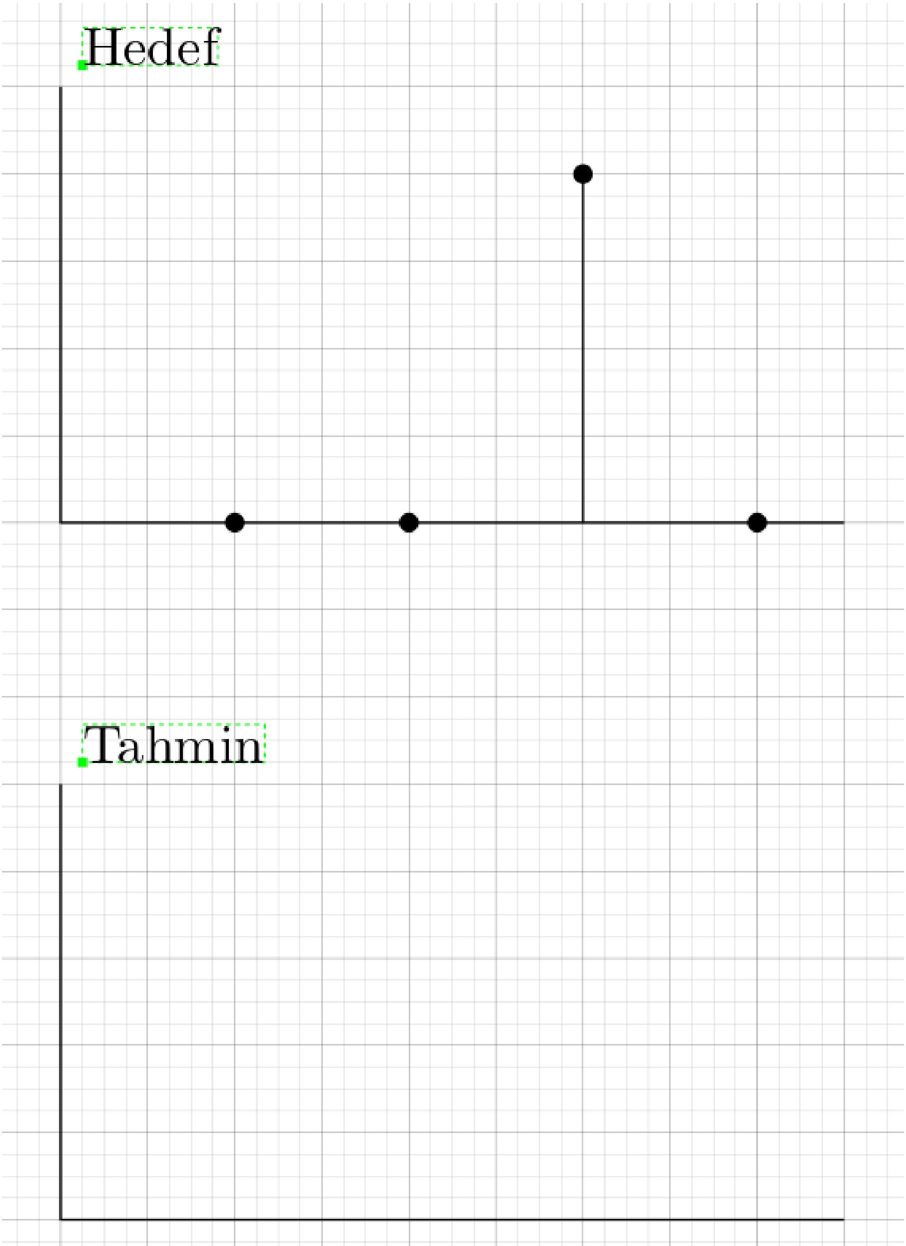


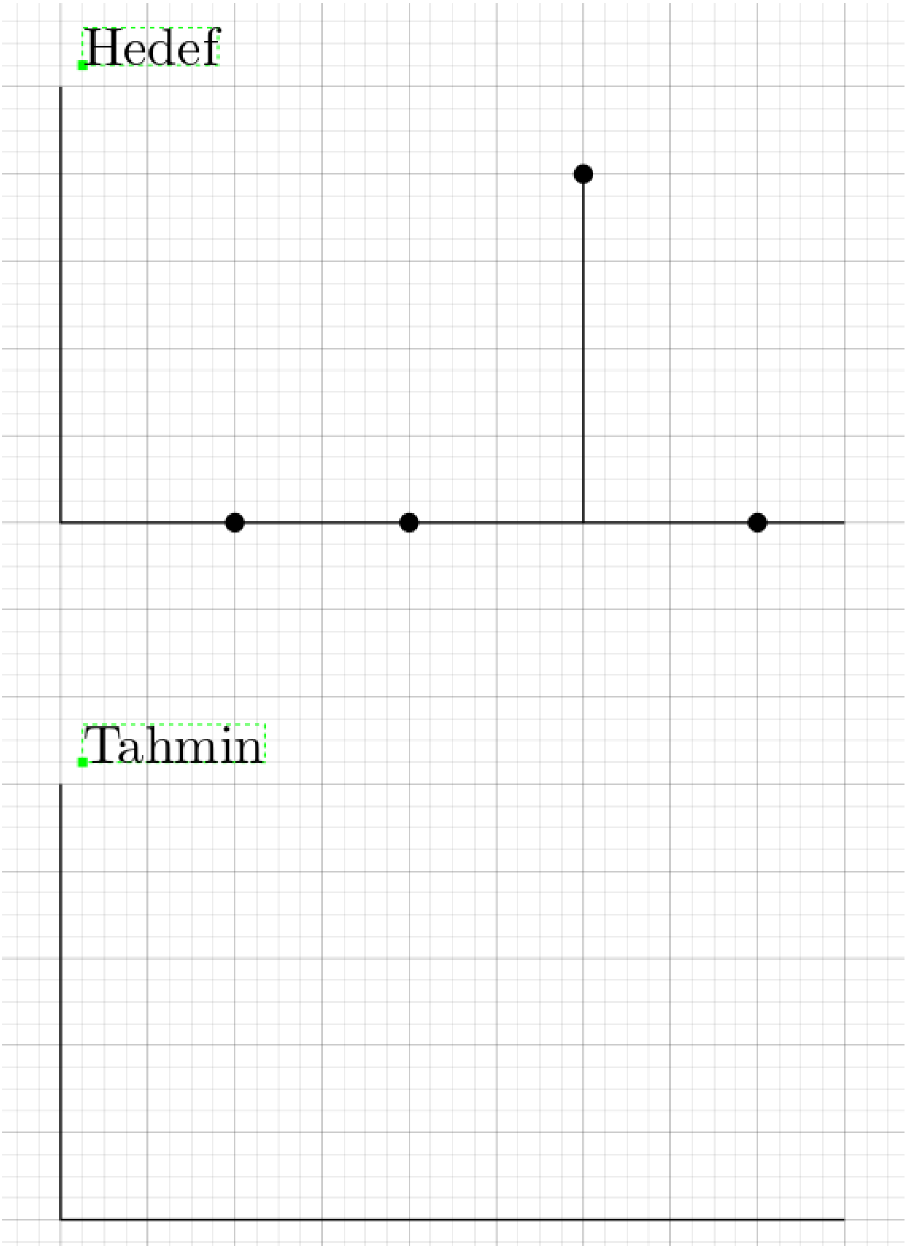
TARGET	PREDICTION
1	0.5
0	0.3
0	0.2

$$\text{Loss for class } X = -\underbrace{p(X)}_{\substack{\text{probability}\\ \text{of class } X\\ \text{in TARGET}}} \cdot \log \underbrace{q(X)}_{\substack{\text{probability}\\ \text{of class } X\\ \text{in PREDICTION}}}$$

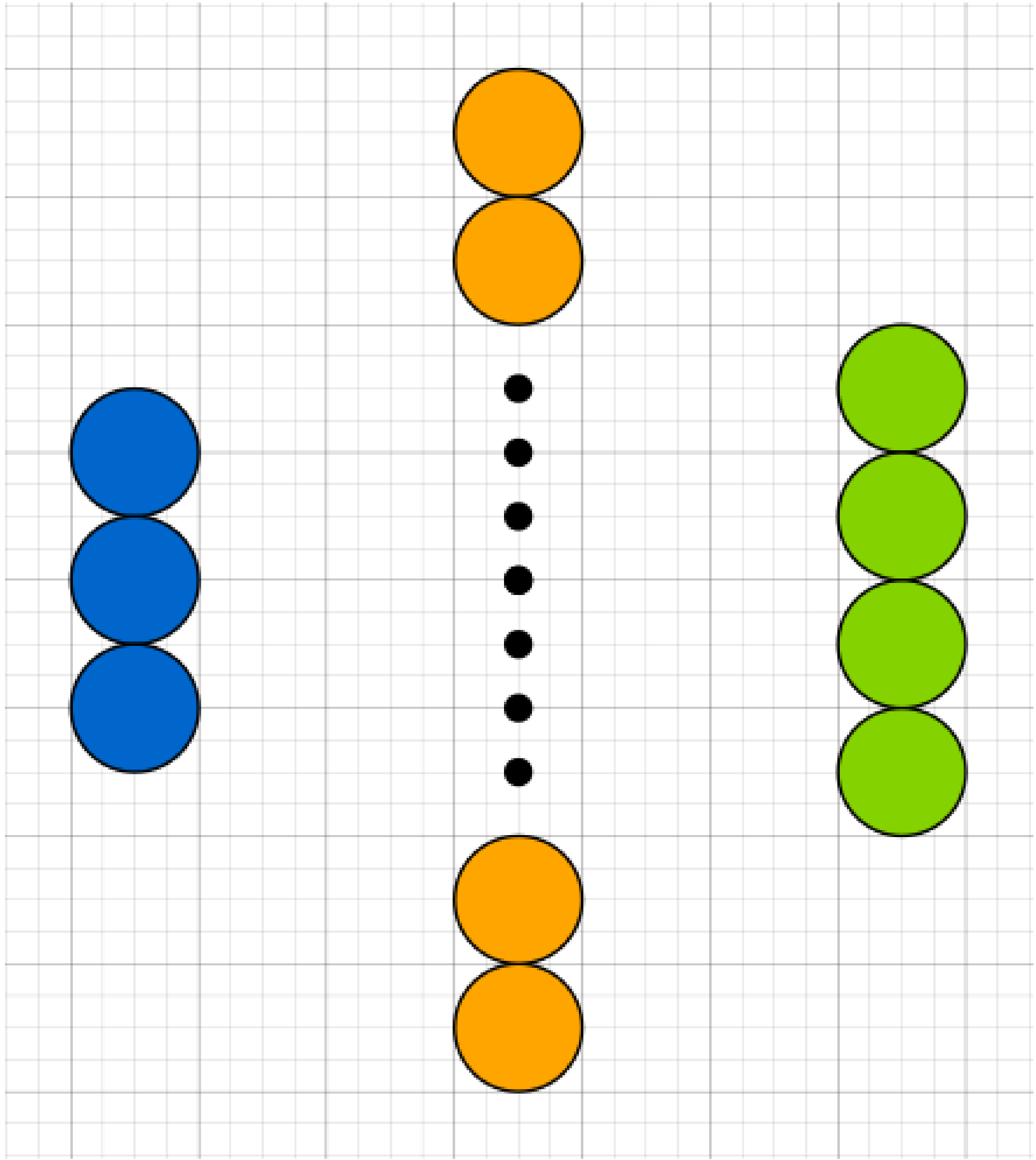
https://towardsdatascience.com/cross-entropy-for-classification-d98e7f974451

Multi Layer Perceptron (MLP)





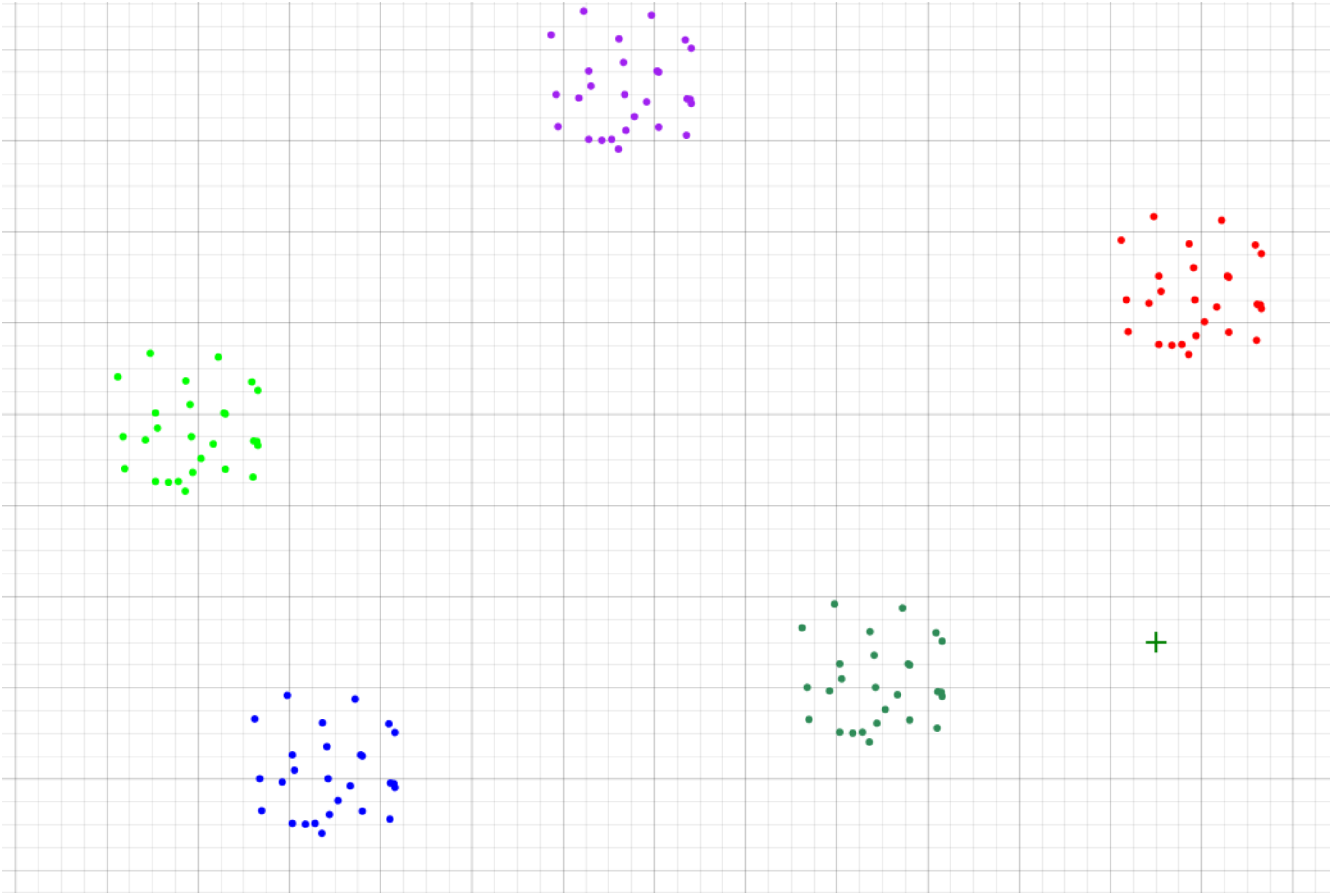
Multi Layer Perceptron (MLP)



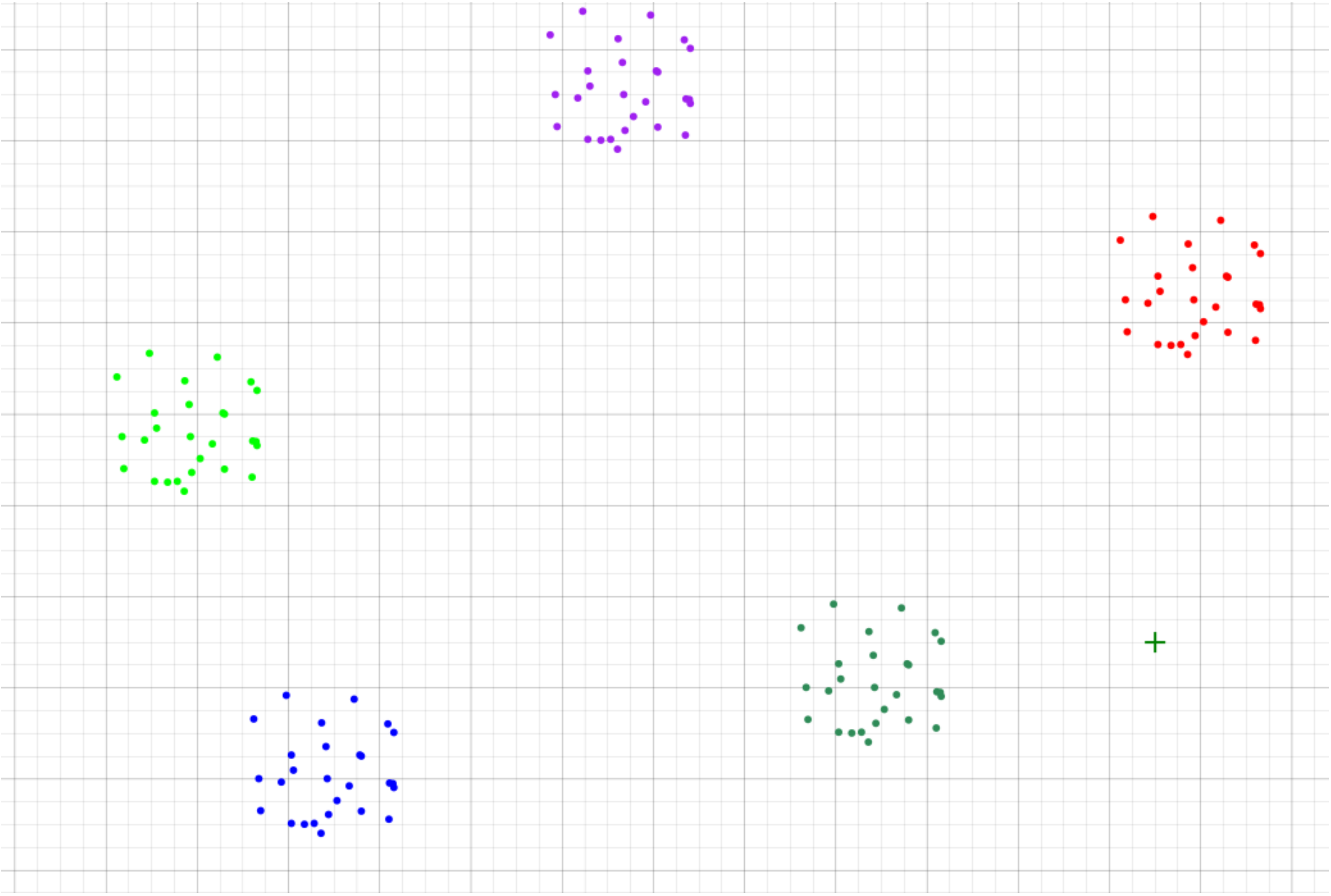
Aktivasyon Fonksiyonları

https://pytorch.org/docs/stable/nn.html#non-linear-activations-other

Yapay Sinir Ağları

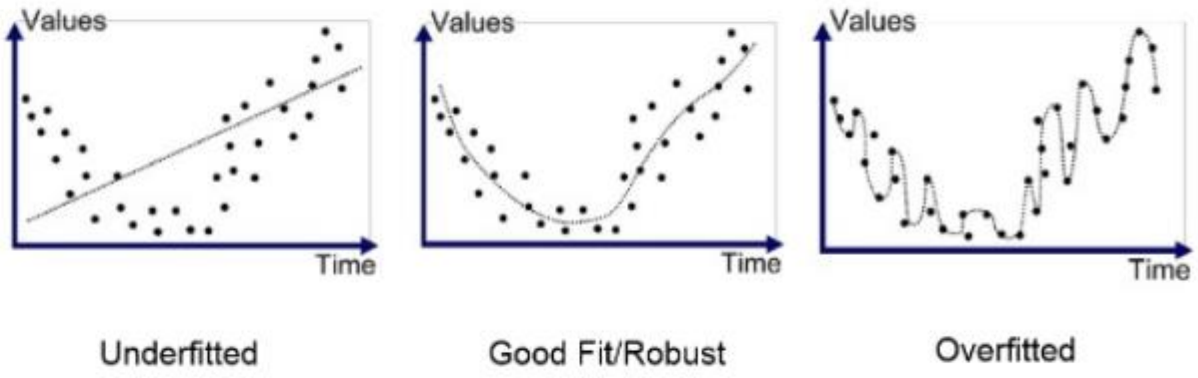


Yapay Sinir Ağları

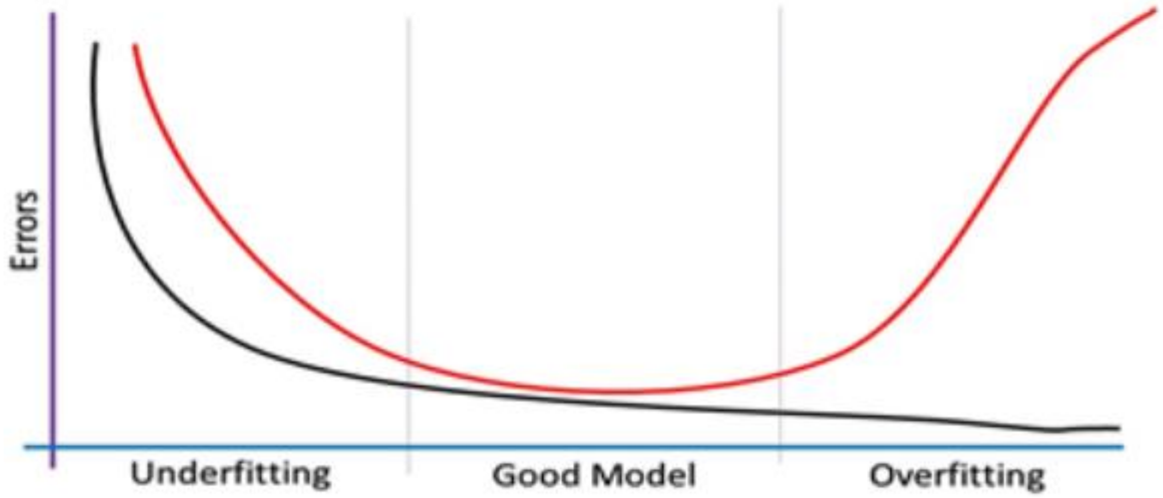


Multi Layer Perceptron (MLP)

Aşırı Öğrenme (Overfitting)



https://medium.com/data-science-tr/overfitting-underfitting-cross-validation-b47dfda0cf4e



https://medium.com/kodluyoruz/makine-%C3%B6%C4%9Frenmesinde-overfitting-underfitting-best-fitting-kavramlar%C4%B1-9f80a5d42719

Multi Layer Perceptron (MLP)

Aşırı Öğrenme (Overfitting)

→ *Daha fazla veri ile eğitim (Training with more data)*: Örnek sayımızı artırmak verimizdeki target ile feature arasında ki ilişkiyi daha rahat anlamamızı sağlamayabilmektedir. Daha fazla veriye sahip olan algoritmanın, daha fazla veri türünü dikkate alarak daha iyi genelleme olasılığı daha yüksektir.

Data augmentation

→ *Regularization*: Düzenleme, modelin daha iyi genelleşmesi için öğrenme algoritmasında küçük değişiklikler yapan bir tekniktir. Bu da modelin görünmeyen verilerdeki performansını artırır. Genellikle L1 ve L2 olmak üzere iki türü bulunmaktadır.

https://medium.com/kodluyoruz/makine-%C3%B6%C4%9Frenmesinde-overfitting-underfitting-best-fitting-kavramlar%C4%B1-9f80a5d42719

Multi Layer Perceptron (MLP)

Aşırı Öğrenme (Overfitting)

→ *Removing Features*: Feature setimizden alakasız feature'ları çıkartarak target-feature ilişkisini daha net bir hale getirebilmekteyiz.

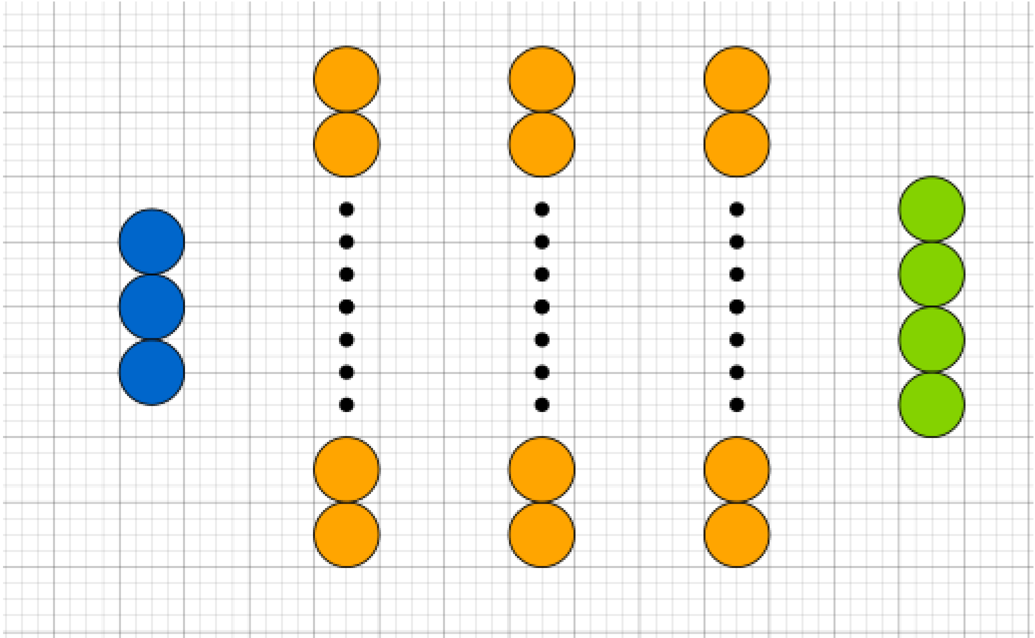
→ *Early stopping:* Çok fazla dönem, eğitim veri kümesinin aşırı sığmasına neden olabilirken, çok azı bir yetersizlik modeliyle sonuçlanabilir. Bir Makine Öğrenimi modelini yinelemeli olarak eğittiğinizde, birkaç yinelemeye kadar modelin performansının arttığını fark edeceksiniz. Belirli bir noktadan sonra, yineleme sayısını artırırsanız, model eğitim veri kümesinde daha iyi performans gösterir, ancak model aşırı yüklenir ve test veri kümelerinde düşük performans gösterir. Bu nedenle, modele fazla uymadan önce modelinizin eğitim tekrarlarını durdurmalısınız.

Nöron sayısını düşürmek

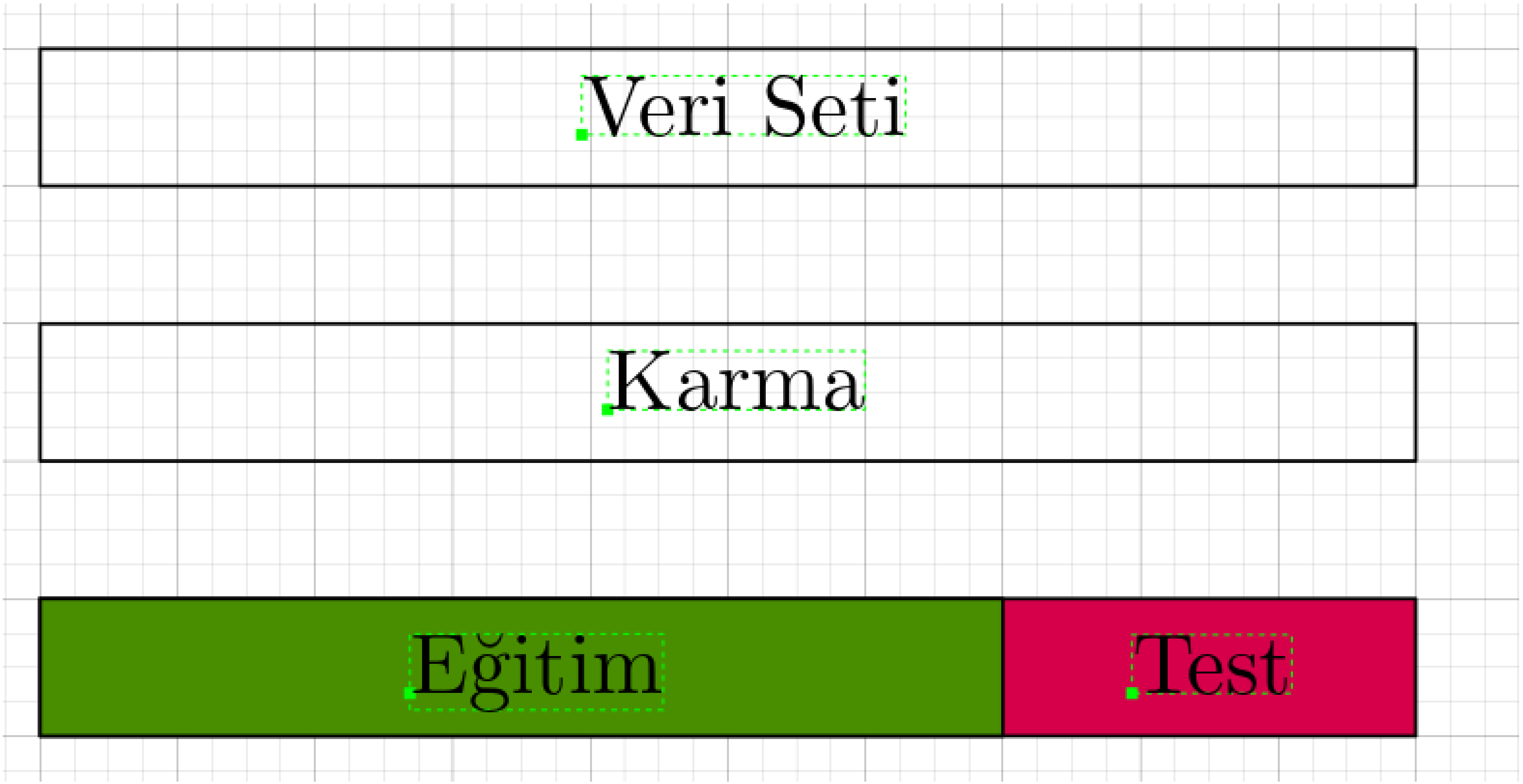
https://medium.com/kodluyoruz/makine-%C3%B6%C4%9Frenmesinde-overfitting-underfitting-best-fitting-kavramlar%C4%B1-9f80a5d42719

Multi Layer Perceptron (MLP)

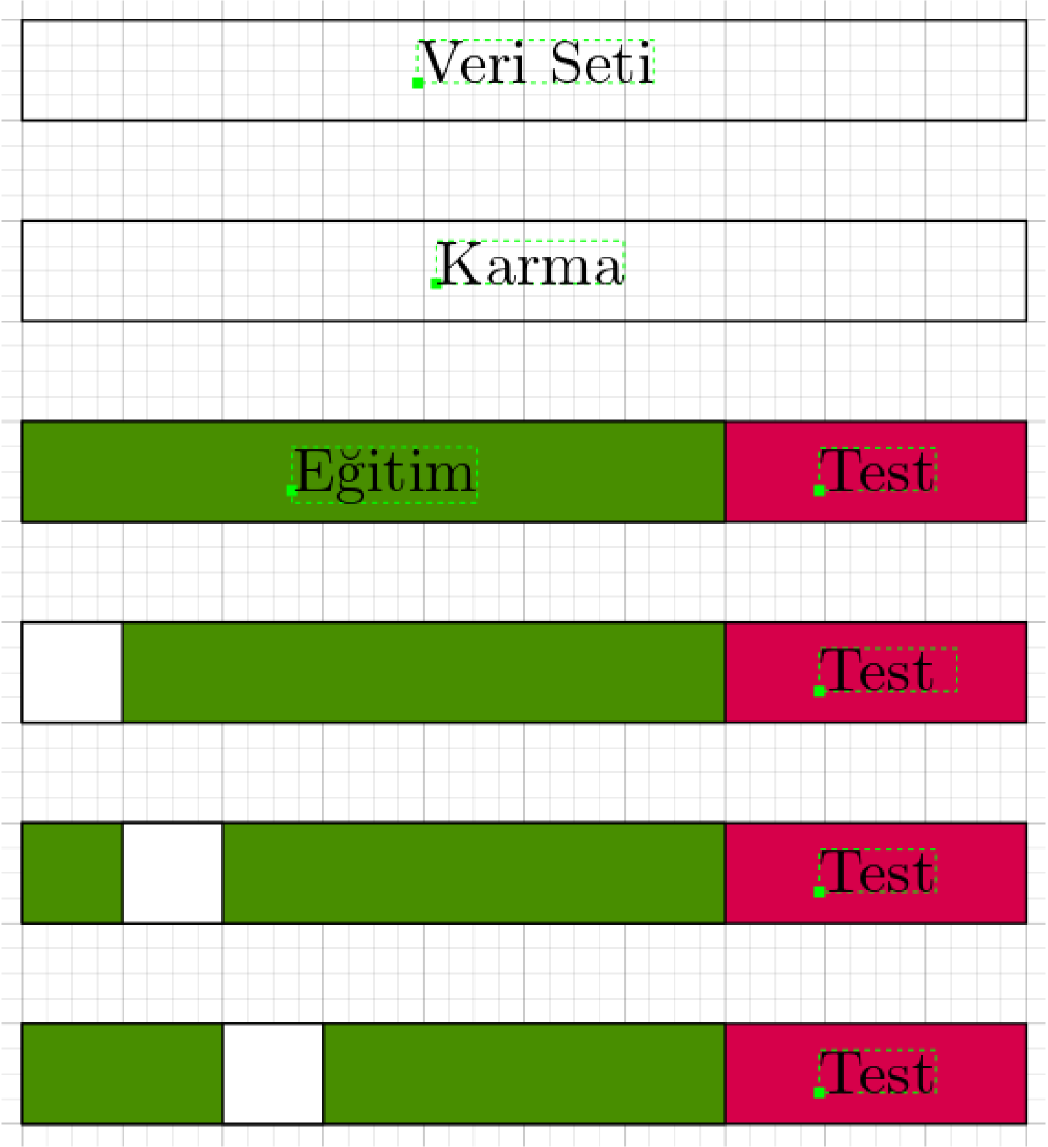
Shallow vs Deep



K-kat Çapraz Doğrulama

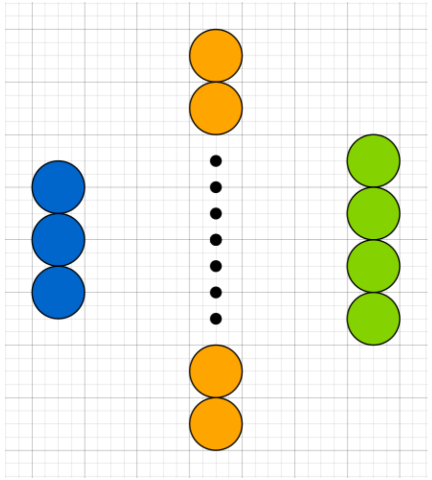


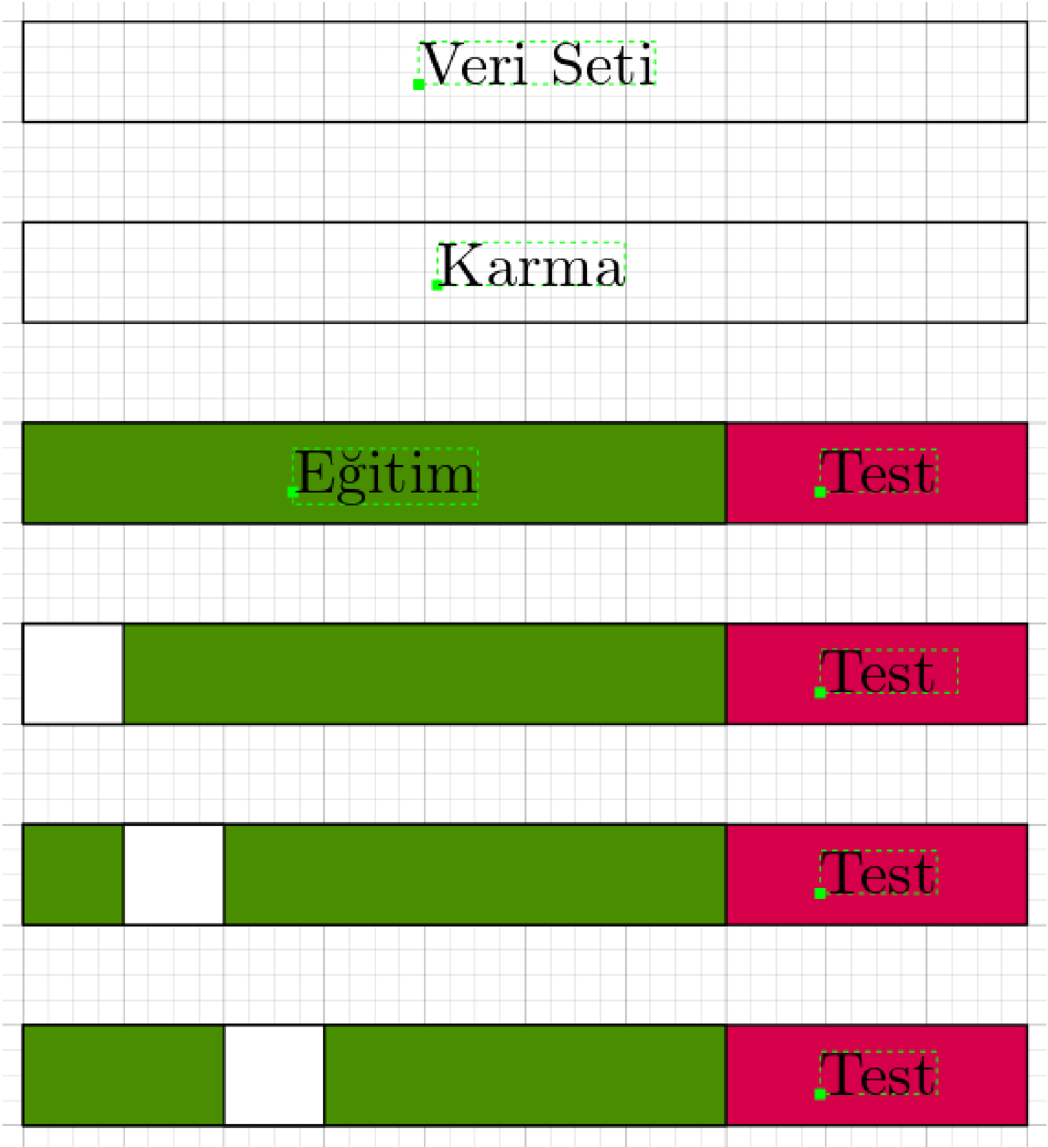
K-kat Çapraz Doğrulama



Multi Layer Perceptron (MLP)

k-Kat Çapraz Doğrulama (k-Fold Cross Validation)





Üst parametre ayarlama

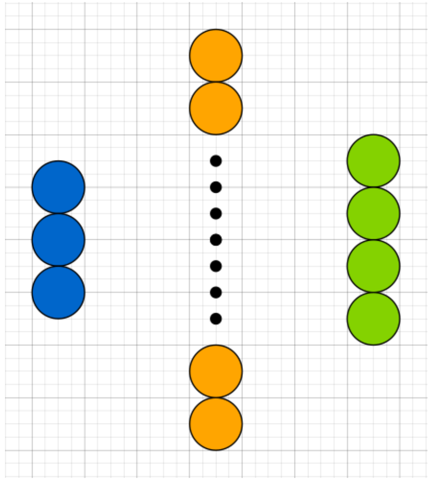
Kaç katman

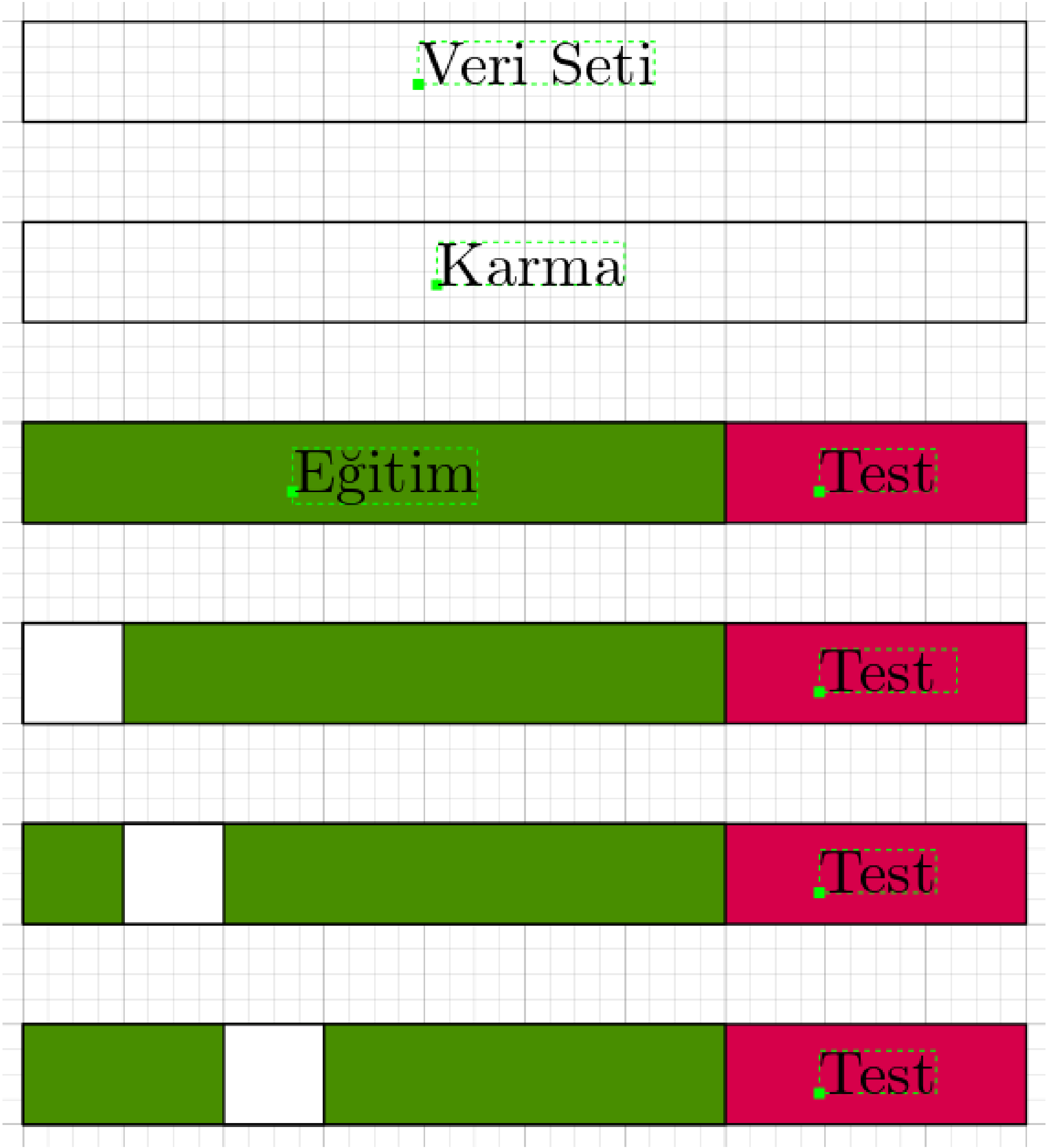
Kaç nöron

vb.

Multi Layer Perceptron (MLP)

k-Kat Çapraz Doğrulama (k-Fold Cross Validation)





- Öğrenme oranı
- «Mini batch» miktarı
- GD algoritması
- İlk öğrenme oranı
- vs. vs.