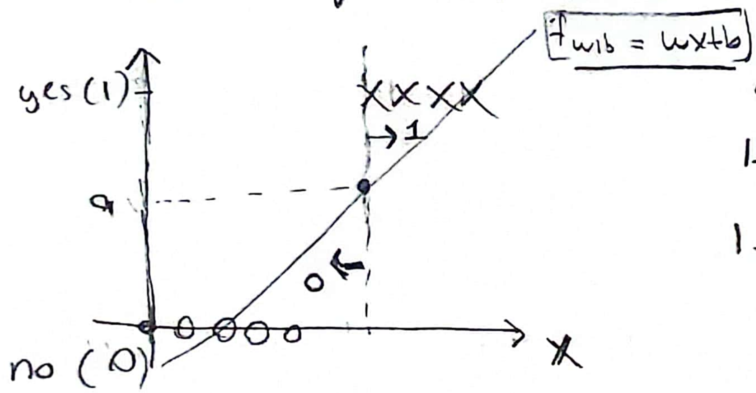
Classification

* y can only be one of two values. This type of classification is called binary classification.



$0 < a < 1$

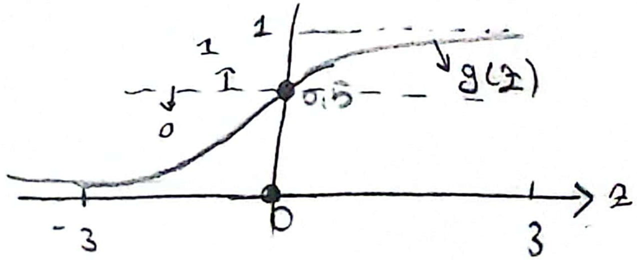
If $f_{w,b}(x) < a \rightarrow \hat{y} = 0$

If $f_{w,b}(x) \geq a \rightarrow \hat{y} = 1$.

Logistic Regression

Logistic regression is used to solve binary classification problems with output label y is either zero or one.

Sigmoid function (logistic function)



* outputs between 0 and 1.

$$g(z) = \frac{1}{1+e^{-z}} \qquad \underline{0 < g(z) < 1}$$

$f_{\vec{w},b}(\vec{x})$

$z = \vec{w}\cdot\vec{x} + b$ " logistic regression

$$f_{\vec{w},b}(\vec{x}) = g(\underbrace{\vec{w}\cdot\vec{x}+b}_{z}) = \boxed{\frac{1}{1+e^{-(\vec{w}\cdot\vec{x}+b)}}}$$

CamScanner

$P(y=0) + P(y=1) = 1$

*Lojistik regresyon, iki veri faktörü arasındaki ilişkileri bulmak için matematikten yararlanan bir veri analizi tekniğidir. Lojistik regresyon, daha sonra diğerine dayalı bu faktörlerden birinin değerini tahmin etmek için bu ilişkiyi kullanır. Tahminin evet ya da hayır gibi sınırlı sonucu vardır.

Lojistik regresyon, yapay zeka ve makine öğrenimi alanında önemli bir tekniktir. Lojistik regresyon kullanılarak oluşturulan ML modelleri, kuruluşların iş verilerinden eyleme dönüştürülebilir öngörüler elde etmelerine yardımcı olur.

Decision Boundary

- $\vec{w}\cdot\vec{x}+b \geq 0$
  $\hat{y} = 1$
- $\vec{w}\cdot\vec{x}+b < 0$
  $\hat{y} = 0$

decision boundary

$$\boxed{z = \vec{w}\cdot\vec{x}+b = 0}$$

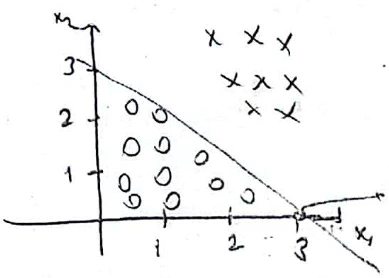
* eşik değerini düşük seçmek yanlış pozitif olasılığını azaltacaktır.

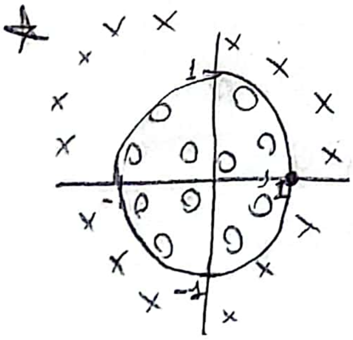
ör//

$z = \vec{w}\cdot\vec{x}+b=0$

$z = x_1 + x_2 - 3 = 0$

$x_1 + x_2 = 3$





non-lineer

$\rightarrow z = w_1 x_1^2 + w_2 x_2^2 + b$

$f_{\vec{w},b}(\vec{x}) = g(z) = g(w_1 \cdot x_1^2 + w_2 \cdot x_2^2 + b)$

$= g(x_1^2 + x_2^2 - 1)$

$x_1^2 + x_2^2 = 1$

$x_1^2 + x_2^2 \geq 1 \rightarrow \hat{y} = 1$

$x_1^2 + x_2^2 < 1 \rightarrow \hat{y} = 0$

Logistic loss function

convex — can reach global min.

$$L(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}) = \begin{cases} -\log(f_{\vec{w},b}(\vec{x}^{(i)})) & \text{if } y^{(i)} = 1 \\ -\log(1 - f_{\vec{w},b}(\vec{x}^{(i)})) & \text{if } y^{(i)} = 0 \end{cases}$$

As $f_{\vec{w},b}(\vec{x}^{(i)}) \rightarrow 1$ then loss $\rightarrow 0$ ✓

As $f_{\vec{w},b}(\vec{x}^{(i)}) \rightarrow 0$ then loss $\rightarrow \infty$ X

* The further prediction $f_{\vec{w},b}(\vec{x}^{(i)})$ is from target $y^{(i)}$, the higher the lo

Cost

$$J(\vec{w},b) = \frac{1}{m} \sum_{i=1}^{m} \underbrace{L\left(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}\right)}_{\text{Loss func.}}$$

* Lojistik regresyon için ortalama karesel hata kullanılıyorsa, maliyet fonk. "konveks değildir", bu nedenle gradyan inişinin w ve b parametreleri için en uygun değeri bulması daha zordur.

$$L\left(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}\right) = -y^{(i)}\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) - (1-y^{(i)})\log\left(1-f_{\vec{w},b}(\vec{x}^{(i)})\right)$$

If $y^{(i)}=1$

* $L\left(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}\right) = -1.\log(f(\vec{x}))$

If $y^{(i)}=0$

* $L\left(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}\right) = -\log\left(1-f_{\vec{w},b}(\vec{x}^{(i)})\right)$

$$J(\vec{w},b) = \frac{-1}{m} \sum_{i=1}^{m} \left[y^{(i)}\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) + (1-y^{(i)})\log\left(1-f_{\vec{w},b}(\vec{x}^{(i)})\right)\right]$$

(maximum likelihood estimation)

Training Logistic Regression

$$w_j = w_j - \alpha \boxed{\frac{\partial}{\partial w_j} J(\vec{w},b)} \rightarrow \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)x_j^{(i)}$$

$$b = b - \alpha \boxed{\frac{\partial}{\partial b} J(\vec{w},b)} \longrightarrow \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)$$

**

Linear regression = $f_{\vec{w},b}(\vec{x}) = \vec{w}\cdot\vec{x} + b$

Logistic regression = $f_{\vec{w},b}(\vec{x}) = \dfrac{1}{1+e^{-(\vec{w}\cdot\vec{x}+b)}}$

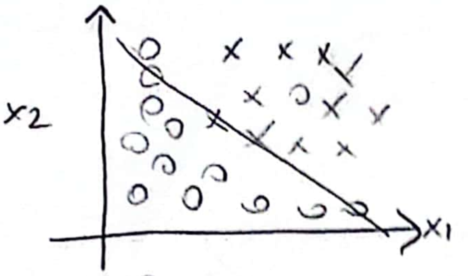
The Problem of Overfitting

- Underfit / high bias doesn't fit the training set well (few features)
- Generalization fits training set pretty well (quadratic features)
- Overfit / high variance fits the training set extremely well (too many features)

* Bir model oluştururken amacımız, yeni örnekler için sonuçları doğru tahmin etmek üzere modeli kullanabilmektir. Bunu yapan bir modelin iyi genelleme yaptığı söylenir.

* Bir model yeni örneklerle iyi çalışmadığında, bu durum modelin iyi genelleme yapmadığını gösterir. (Aşırı uyum)

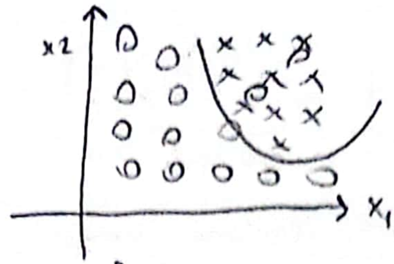
CamScanner



$z = w_1x_1 + w_2x_2 + b$

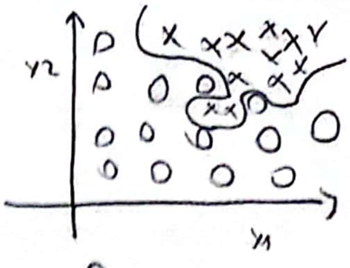
$f_{\vec{w},b}(\vec{x}) = g(z)$

(underfit / high bias)



$z = w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_2^2 + w_5x_1x_2 + b$

(just right)



$z = w_1x_1 + w_2x_2 + w_3x_1^2 x_2 + \dots + b$

(overfit)

Regularization (Düzenlileştirme)

Değişken sayısı arttıkça, modelin aşırı öğrenme (overfit) olasılığı artmaktadır. Düzenlileştirme de aşırı öğrenme problemini çözmek için kullanılan bir tekniktir.

* Regularization lets you keep all of your features, but they just prevents the features from having an overly large effect, which is what sometimes can cause overfitting.

* w_j parametrelerini düzenlileştiririz, b parametresini düzenlemek çok önemli değil.

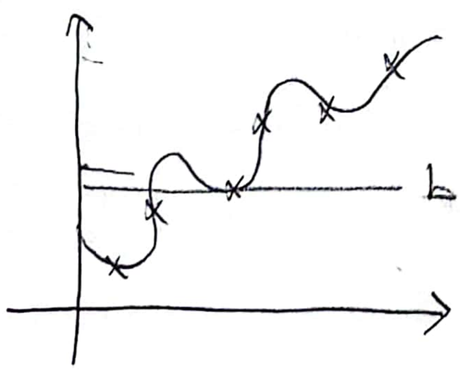
Addressing overfitting

1. collect more data
2. select features
   - Feature selection
3. Reduce size of parameters
   - "Regularization"

Düzenli hale getirme uygulamak, eğitim örneklerinin sayısını arttırmak veya en alakalı özelliklerin bir alt kümesini seçmek gibi yöntemler, modelin eğitim setinde olmayan yeni örneklere daha iyi genelleme yapılmasına yardımcı olur.

$$J(\vec{w},b) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)^2 + \underbrace{\frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2}_{\text{regularization term}} + \underbrace{\frac{\lambda}{2m} b^2}_{\text{can include or exclude } b}\right]$$

regularization parameter $\lambda > 0$



if λ is very large, $\lambda = 10^{10}$
the learning algorithm will choose w parameters are close to 0 and $F(x) = b$. So the learning algorithm fits a horizontal straight line and under fits.

$$f_{\vec{w},b}(\vec{x}) = \underset{0}{\cancel{w_1}}x + \underset{0}{\cancel{w_2}}x^2 + \underset{0}{\cancel{w_3}}x^3 + \underset{0}{\cancel{w_4}}x^4 + b$$

$$\boxed{f(x) = b}$$

Regularized Linear Regression

$$\min_{\vec{w},b} J(\vec{w},b) = \min_{\vec{w},b}\left(\frac{1}{2m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)})-y^{(i)}\right)^2 + \frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2\right)$$

$$w_j = w_j - \alpha \boxed{\frac{\partial}{\partial w_j} J(w_j,b)} \to \left[\frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)})-y^{(i)}\right)x_j^{(i)}\right] + \frac{\lambda}{m} w_j$$

$$b = b - \alpha \boxed{\frac{\partial}{\partial b} J(w_j,b)} \to \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)})-y^{(i)}\right)$$