

# Hata Analizi -2

Prof. Dr. M. Elif Karslıgil  
Yıldız Teknik Üniversitesi  
Computer Engineering Department  
Intelligent Systems Lab.

# *Loss Function Nedir ?*

Bir ML modelinin temel amacı, modelin hata payını en aza indirmektir.

- **Loss Function:** Tek bir veri örneği (örn. tek bir fotoğraf) için hesaplanan hatadır.
- **Cost Function:** Tüm eğitim setindeki (tüm veri) hataların ortalamasıdır.
- **Düşük Kayıp (Low Loss):** Modelin tahminleri gerçeğe çok yakındır.
- **Yüksek Kayıp (High Loss):** Modelin daha çok öğrenmesi gereklidir.

# *Regresyon Problemleri için Loss Function*

- **Mean Squared Error (MSE):** Büyük hatalar çok daha ağır cezalandırır.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

- **Mean Absolute Error (MAE):** Outlier değerlere karşı daha dirençlidir.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

# *Regresyon Problemleri için Loss Function*

Bir ev fiyatı tahmin modeli 3 örnek için aşağıdaki tahminleri yapmıştır.

Ev No	Gerçek Fiyat (y)	Modelin Tahmini ( $y^$ )
1. Ev	100.000 TL	110.000 TL
2. Ev	200.000 TL	190.000 TL
3. Ev	150.000 TL	150.000 TL

$$MSE = 66.6$$

$$MAE = 6.6$$

# *Sınıflandırma için Loss Function*

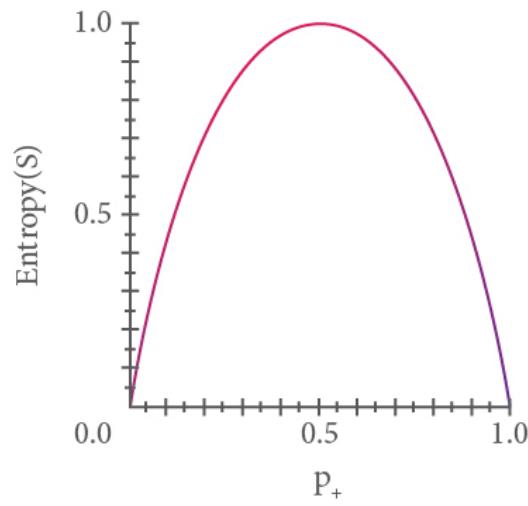
E-posta spam mi değil mi? Resimdeki hayvan nedir?

- **Binary Cross-Entropy (Log Loss):** Sadece iki seçenek olduğunda kullanılır. Modelin tahmini (olasılık değeri) doğrudan gerçek sınıfla karşılaştırılır.
- **Categorical Cross-Entropy:** İkiden fazla sınıf olduğunda kullanılır.

# Entropy

- Entropy S örneklerinin saflığını ölçer

İkili sınıflandırma:

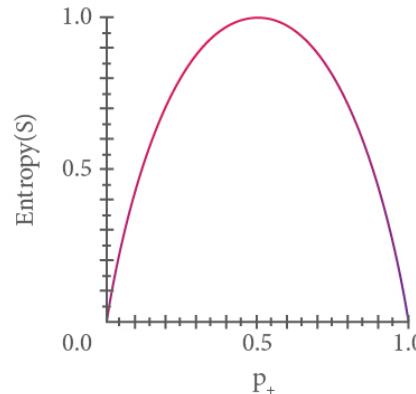
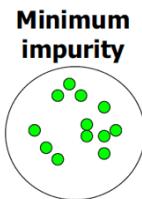


- $S$  : eğitim örnekleri
- $p_+$  pozitif örneklerin olasılığı
- $p_-$  negatif örneklerin olasılığı
- Entropy + veya – sınıfları kodlamak için beklenen bit sayısı

$$\text{Entropy} = \sum_i -p_i \log_2 p_i$$

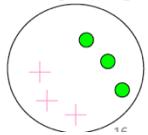
# Entropy

$$\text{Entropy} = - \sum_i p_i \log_2 p_i$$



- Bütün örnekler aynı sınıftaysa Entropy = 0  
entropy = - 1 log<sub>2</sub>1 = 0

Maximum impurity



- Örnekler yarı yarıya dağılmışsa Entropy = 1  
entropy = -0.5 log<sub>2</sub>0.5 – 0.5 log<sub>2</sub>0.5 =1

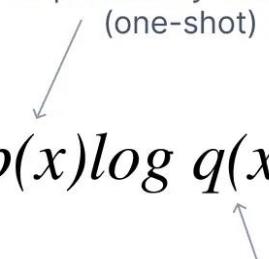
# Cross Entropy Loss Function

- İki ayrı olasılık dağılımı arasındaki *cross entropy*, iki dağılımın ne kadar benzer olduğunu gösterir.
- Tahmin edilen çıkış ile beklenen çıkış arasındaki Cross Entropy

$$H(p, q) = - \sum_{x \in \text{classes}} p(x) \log q(x)$$

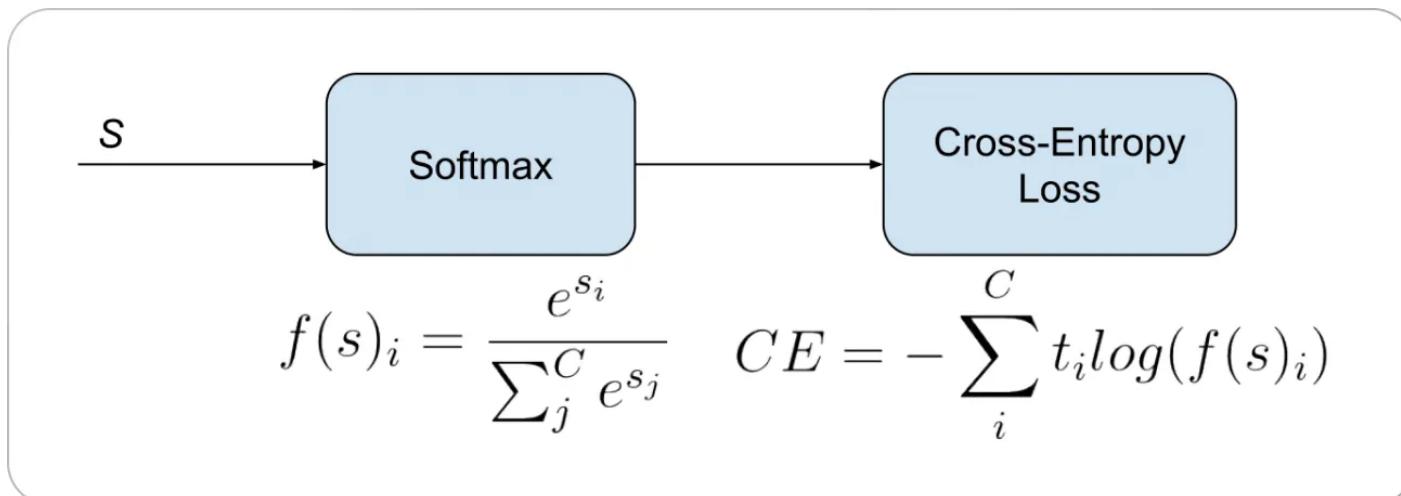
True probability distribution  
(one-shot)

Your model's predicted  
probability distribution

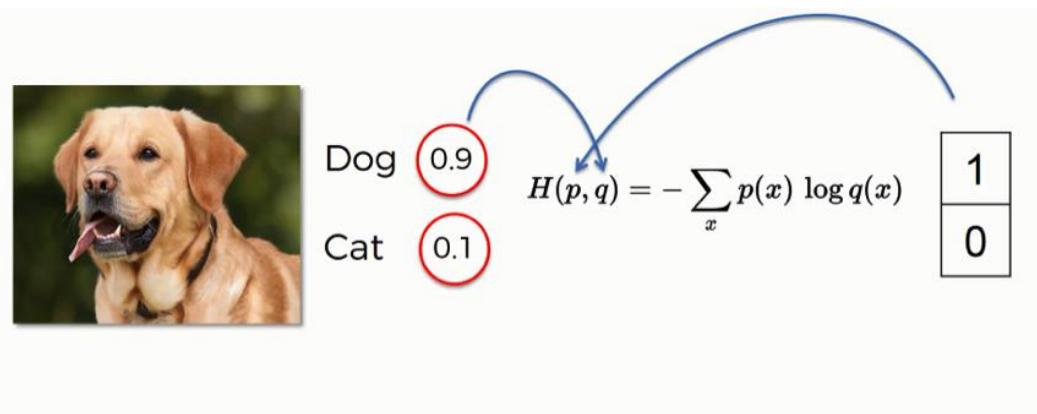
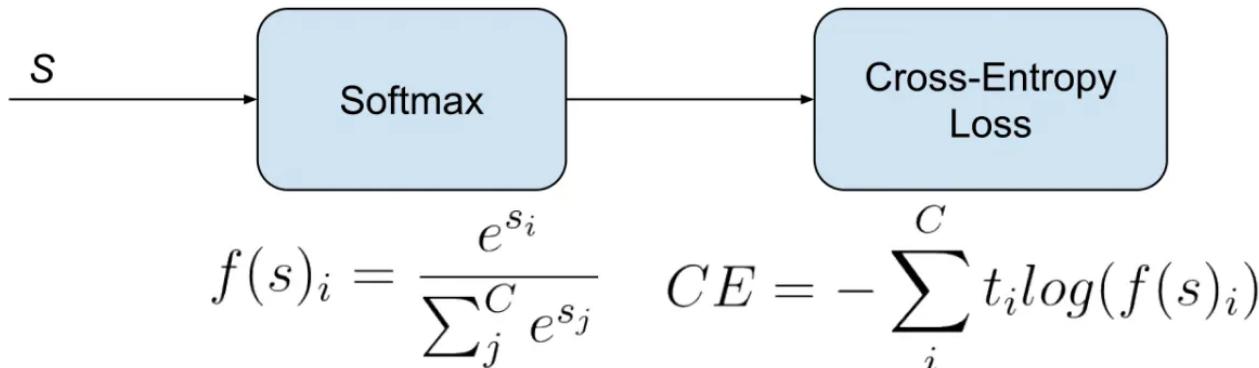


# Cross Entropy Loss Function

- Eğitim sırasında ağırlıklar güncellenirken kullanılır
- Hesaplanan çıkış ile beklenen çıkış arasındaki mesafeyi hesaplar.
- Mükemmel modelde cross-entropy loss sıfır olur.
- CNN için CE hesabı:



# Cross Entropy Loss Function



# Softmax and Cross Entropy

- to train the network with backpropagation, we calculate the derivative of the loss :

$$\frac{\partial L}{\partial l_n} = \hat{y}_n - y_n$$

The derivative of the loss with respect to the n-th node in the last hidden layer...

...is the n-th component of the network's prediction...

...minus the n-th component of the label.

# Binary Cross Entropy (Log Loss)

İki sınıflı bir problemde tek örnek için hata fonksiyonu:

$$Loss = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$$

**Örnek:** Bir e-postanın **Spam (1)** olup olmadığını tahmin ediyoruz.

- Gerçek Durum:** E-posta Spam ( $y=1$ ).
- Modelin Tahmini:** %80 ihtimalle Spam ( $y'=0.8$ ).

$$Loss = -[1 \cdot \log(0.8) + (1 - 1) \cdot \log(1 - 0.8)]$$

$$Loss = -[\log(0.8) + 0]$$

$$Loss \approx -(-0.097) = 0.097$$

# Binary Cross Entropy (Log Loss)

- N adet örnek için toplam loss:

$$J(\theta) = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

Örnek	Gerçek Durum ( $y$ )	Model Tahmini ( $\hat{y}$ )	Loss Hesabı ( $-[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$ )
1. Veri	1 (Dolandırıcı)	0.90	$-\log(0.90) \approx \mathbf{0.105}$
2. Veri	0 (Normal)	0.10	$-\log(0.90) \approx \mathbf{0.105}$
3. Veri	1 (Dolandırıcı)	0.40	$-\log(0.40) \approx \mathbf{0.916}$

**Toplam Kayıp:**  $0.105 + 0.105 + 0.916 = 1.126$

**Toplam Maliyet(cost):**  $1.1126 / 3 = 0.375$

# Binary Cross Entropy

Öğrenmede türev alan(gradyan tabanlı) yöntemler kullanır.

- BCE kullanan temel ML yöntemleri:

Lojistik Regresyon

Olasılıksal Destek Vektör Makineleri (Son katmanda lojistik reg. var)

XGBoost, LightGBM, CatBoost

- BCE kullanan derin öğrenme tabanlı yöntemler:

**MLP** (İkili sınıflandırma yapan sinir ağları modelleri)

**CNN** (Bu görüntüde kedi var mı? )

**RNN/LSTM** (Olumlu/Olumsuz Duygu analizi)

**Autoencoder**

**GAN** (Görüntü gerçek mi değil mi)

# Categorical Cross Entropy

- Üç veya daha fazla sınıfın bulunduğu (Multi-class Classification) problemlerde kullanılır.
- Modelin çıktı olarak verdiği olasılık dağılımı ile verinin gerçek dağılımı arasındaki farkı (entropiyi) ölçer.
- CCE kullanabilmek için modelin son katmanında iki şart vardır:
  - 1.Nöron Sayısı: Sınıf sayısı kadar nöron olmalıdır.
  - 2.Aktivasyon: Çıktılar Softmax fonksiyonundan geçmelidir.

# Categorical Cross Entropy

N adet veri örneği ve C adet sınıf için toplam maliyet (Cost):

$$J = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C y_{i,j} \log(\hat{y}_{i,j})$$

- $y_{i,j}$ : i. örneğin j. sınıfına ait olup olmadığı (1 veya 0 - One-Hot Encoding).
- $\hat{y}_{i,j}$ : Modelin i. örneğin j. sınıfına ait olma olasılığı tahmini.

# Categorical Cross Entropy

**Senaryo:** Bir "Köpek" fotoğrafı sisteme girdi.

1. **Gerçek Etiket (One-Hot Encoding):** Köpek sınıfı 2. sırada olsun.  $y = [0, 1, 0]$
2. **Modelin Tahmini (Softmax Çıktısı):** Modelimiz şöyle bir tahminde bulundu:  $\hat{y} = [0.1, 0.7, 0.2]$  (Kedi: %10, Köpek: %70, Kuş: %20)

**Hesaplama:**

Formüldeki iç toplamı ( $\sum y \cdot \log(\hat{y})$ ) her sınıf için uygulayalım:

$$Loss = -((y_{kedi} \cdot \log(\hat{y}_{kedi})) + (y_{köpek} \cdot \log(\hat{y}_{köpek})) + (y_{kuş} \cdot \log(\hat{y}_{kuş})))$$

Değerleri yerine koyalım:

$$Loss = -((0 \cdot \log(0.1)) + (1 \cdot \log(0.7)) + (0 \cdot \log(0.2)))$$

**Sihirli Nokta:** Sıfır ile çarpılan tüm terimler yok olur! Geriye sadece gerçek sınıfa ait olan terim kalır:

$$Loss = -\log(0.7) \approx \mathbf{0.356}$$

# Categorical Cross Entropy Kullanan Modeller

- **Gradyan Tabanlı Gelişmiş Ağaç Modelleri:**
  - XGBOOST, LIGHTGBM, CATBOOST
  - Gradient Boosting Machine
- **Derin Öğrenme ve Yapay Sinir Ağları:**
  - Multilayered Perceptron
  - Evrişimli Sinir Ağları (CNN)
  - Doğal Dil İşleme (NLP) Modelleri: Transformer, BERT

# Loss Function Kullanımı - Özeti

Problem Türü	Örnek Senaryo	Çıkış Katmanı (Nöron + Aktivasyon)	Kayıp Fonksiyonu (Loss)	Temel Mantık
<b>Regresyon</b>	Ev Fiyatı Tahmini	1 Nöron + Lineer (Aktivasyonsuz)	<b>MSE</b> (Mean Squared Error)	Tahmin ile gerçek arasındaki mesafenin karesini alır.
<b>Regresyon</b>	Aykırı Değerli Tahmin	1 Nöron + Lineer	<b>MAE</b> (Mean Absolute Error)	Hataların mutlak değerini alır, aykırı değerlere hoşgörülüdür.
<b>İkili Sınıflandırma</b>	Spam mı? (0 veya 1)	1 Nöron + Sigmoid	<b>Binary Cross-Entropy</b>	Doğru sınıfın olasılığını (0-1 arası) maksimize eder.
<b>Çoklu Sınıflandırma</b>	Kedi, Köpek, Kuş?	Sınıf Sayısı kadar Nöron + <b>Softmax</b>	<b>Categorical Cross-Entropy</b>	Sınıf olasılık dağılımları arasındaki farkı ölçer.