**DEEP LEARNING**

### [**General [Genel]**](https://lms.clarusway.com/course/view.php?id=44#section-0)

In this course you will create and train deep neural networks, work on critical architecture parameters, and apply neural networks and deep learning to applications. [Bu kursta derin sinir ağları oluşturacak ve eğitecek, kritik mimari parametreler üzerinde çalışacak ve sinir ağlarını ve derin öğrenmeyi uygulamalara uygulayacaksınız.]

This course will show you how to design artificial neural networks for deep learning using Google's TensorFlow  framework. [Bu kurs, Google'ın TensorFlow çerçevesini kullanarak derin öğrenme için yapay sinir ağlarının nasıl tasarlanacağını gösterecektir.]  You'll focus on learning about the TensorFlow and it's API Keras to quickly and simply build models. [Modelleri hızlı ve basit bir şekilde oluşturmak için TensorFlow ve API Keras'ı öğrenmeye odaklanacaksınız.]

Building a Convolutional Neural Network and applying it to detection and recognition tasks is another exiting part. [Bir Evrişimli Sinir Ağı oluşturmak ve bunu algılama ve tanıma görevlerine uygulamak, bir diğer heyecan verici kısımdır.] Finally implementing Recurrent Neural Network and some doing some basic applications of it will be the last subjects in this course. [Son olarak Recurrent Neural Network'ün uygulanması ve bazı temel uygulamalarının yapılması bu dersin son konuları olacaktır.]

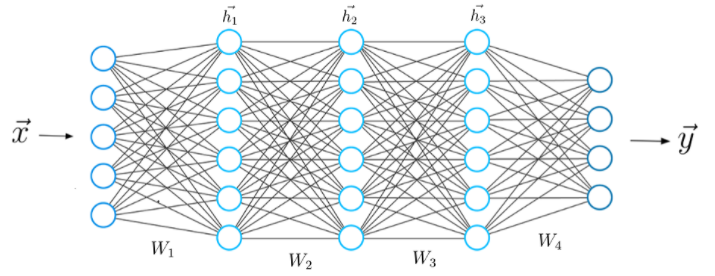
# **But what is a neural network? | Chapter 1, Deep learning**

[**https://www.youtube.com/watch?v=aircAruvnKk**](https://www.youtube.com/watch?v=aircAruvnKk)

### **Basic Concepts in Deep Learning**

**Introduction**

Welcome to Deep Learning Course.

****

In this section of the course, you will learn the basic concepts of Deep Learning with Artificial Neural Networks in the following order: [Kursun bu bölümünde, Yapay Sinir Ağları ile Derin Öğrenmenin temel kavramlarını aşağıdaki sırayla öğreneceksiniz:]

* Neural Networks [Nöral ağlar]
* Activation Functions [Aktivasyon Fonksiyonları]
* Cost Functions [Maliyet Fonksiyonları]
* Gradient Descent [Dereceli alçalma]
* Backpropagation. [Geri yayılım.]

Then, you will have a chance to study regression and classification problems with Deep Learning. [Ardından, Derin Öğrenme ile regresyon ve sınıflandırma problemlerini inceleme şansınız olacak.]  
  
After finishing ANN subjects we will cover the Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) sections. [YSA konularını bitirdikten sonra Evrişimsel Sinir Ağları (CNN) ve Tekrarlayan Sinir Ağları (RNN) bölümlerini işleyeceğiz.] Using CNN you will solve classification problems and using RNN you will solve time series problems. [CNN kullanarak sınıflandırma problemlerini çözeceksiniz ve RNN kullanarak zaman serisi problemlerini çözeceksiniz.]

# **What is Deep Learning?**

<https://www.youtube.com/watch?v=BRGzC2nuO6U&t=1s>

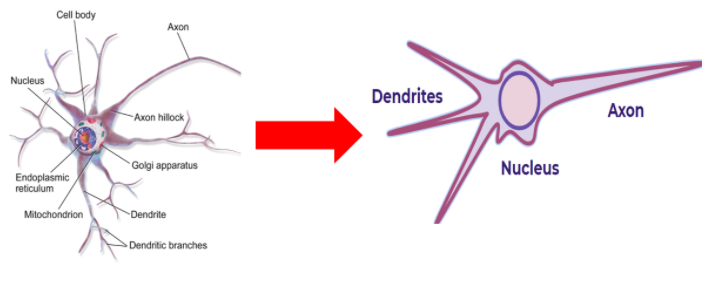
## **Perceptron Model and Neural Networks**

### **Perceptron Model**

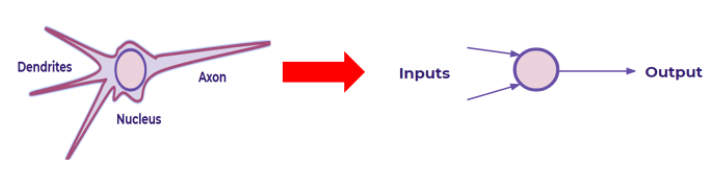
In this lesson, you will learn the following abstractions in Deep Learning. [Bu derste, Derin Öğrenmede aşağıdaki soyutlamaları öğreneceksiniz.]

* Single Biological Neuron [Tek Biyolojik Nöron]
* Perceptron Model [Algılayıcı modeli]
* The Function in the Perceptron Model [Perceptron Modelindeki İşlev]

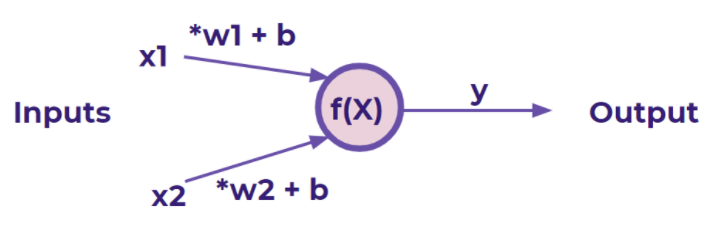
### 1. [1.] Single Biological Neuron [Tek Biyolojik Nöron] Since the whole idea behind deep learning is to have computers artificially mimic biological natural intelligence, you will start by the understanding of how biological neurons work. [Derin öğrenmenin arkasındaki tüm fikir, bilgisayarların biyolojik doğal zekayı yapay olarak taklit etmesi olduğundan, biyolojik nöronların nasıl çalıştığını anlayarak başlayacaksınız.] The picture below shows how to simplify an actual biological neuron. [Aşağıdaki resim gerçek bir biyolojik nöronun nasıl basitleştirileceğini göstermektedir.]



**2. [2.] Perceptron Model [Algılayıcı modeli]**  
The idea behind the perceptron model is, converting a simplified biological neuron to a mathematical model. [Algılayıcı modelinin arkasındaki fikir, basitleştirilmiş bir biyolojik nöronu matematiksel bir modele dönüştürmektir.] The history of the perceptron model dates back to 1958. [Perceptron modelinin tarihi 1958 yılına kadar uzanmaktadır.]  The picture below shows how to convert the simplified biological neuron to a perceptron model. [Aşağıdaki resim, basitleştirilmiş biyolojik nöronun bir algılayıcı modeline nasıl dönüştürüleceğini göstermektedir.]



**3. [3.] The Function in the Perceptron Model [Perceptron Modelindeki İşlev]**  
There is a function in the perceptron model that makes the calculations in the model. [Perceptron modelinde, modelde hesaplamaları yapan bir fonksiyon vardır.] In this lecture, you will learn how to create the function that runs inside the perceptron model. [Bu derste, perceptron modeli içinde çalışan fonksiyonun nasıl oluşturulacağını öğreneceksiniz.]



**y=(x1\*w1+b) + (x2\*w2+b)**

where:  
x1, x2 : inputs,  
w1, w2 : weights,  
b : bias.

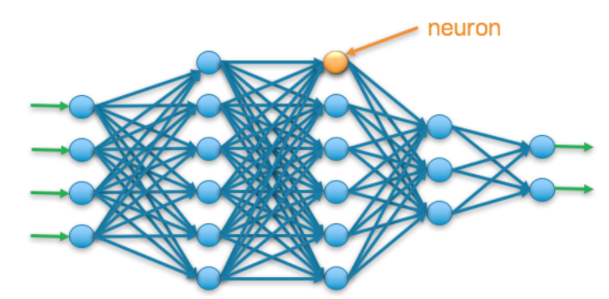
Q: What is deep learning? [S: Derin öğrenme nedir?]  
A: Deep learning is a part of machine learning with an algorithm inspired by the structure and function of the brain, which is called an artificial neural network. [C: Derin öğrenme, yapay sinir ağı olarak adlandırılan beynin yapısından ve işlevinden ilham alan bir algoritma ile makine öğreniminin bir parçasıdır.] In the mid-1960s, Alexey Grigorevich Ivakhnenko published the first general, while working on deep learning network. [1960'ların ortalarında, Alexey Grigorevich Ivakhnenko, derin öğrenme ağı üzerinde çalışırken ilk generali yayınladı.] Deep learning is suited over a range of fields such as computer vision, speech recognition, natural language processing, etc. [Derin öğrenme, bilgisayarla görme, konuşma tanıma, doğal dil işleme vb. gibi bir dizi alan için uygundur.]  
- Interview Q&A

### **Neural Networks [Nöral ağlar]**

So far, you have learned what the perceptron model is. [Şimdiye kadar algılayıcı modelinin ne olduğunu öğrendiniz.] In this lesson, you will learn the following abstractions: [Bu derste, aşağıdaki soyutlamaları öğreneceksiniz:]

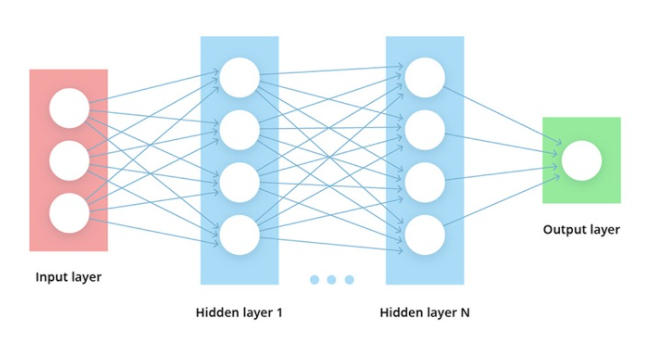
* Neuron [Nöron]
* Layers: Input Layer, Hidden Layer, Output Layer [Katmanlar: Girdi Katmanı, Gizli Katman, Çıktı Katmanı]
* Deep Neural Network [Derin Sinir Ağı]

**1. [1.] Neuron [Nöron]**  
It exactly defines a single perceptron model. [Tam olarak tek bir algılayıcı modelini tanımlar.] After this lesson, we will call the **single** **perceptrons** as **neurons. [Bu dersten sonra, tek algılayıcıları nöronlar olarak adlandıracağız.]**In the image below, each **circle** represents a neuron. [Aşağıdaki resimde her daire bir nöronu temsil etmektedir.]

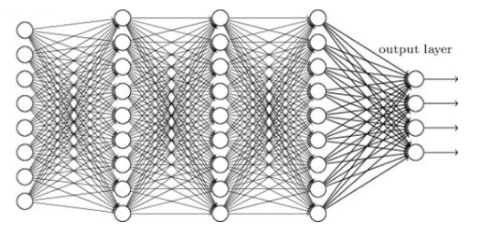


**2. [2.] Layers [Katmanlar]**  
  
You will learn the following three types of layers. [Aşağıdaki üç katman türünü öğreneceksiniz.]

* **Input Layer:** Receives the information [Giriş Katmanı: Bilgileri alır]
* **Hidden Layer:** Where the computations appear [Gizli Katman: Hesaplamaların göründüğü yer]
* **Output Layer:** Where the results of the computations will show up [Çıktı Katmanı: Hesaplamaların sonuçlarının görüneceği yer]



**3. [3.] Deep Neural Network [Derin Sinir Ağı]**  
A neural network is called a deep neural network if it contains two or more hidden layers. [Bir sinir ağı, iki veya daha fazla gizli katman içeriyorsa derin sinir ağı olarak adlandırılır.]



Q: How many layers are there in the neural network? [S: Sinir ağında kaç katman var?]  
A: There are three types of layers: [C: Üç tür katman vardır:]

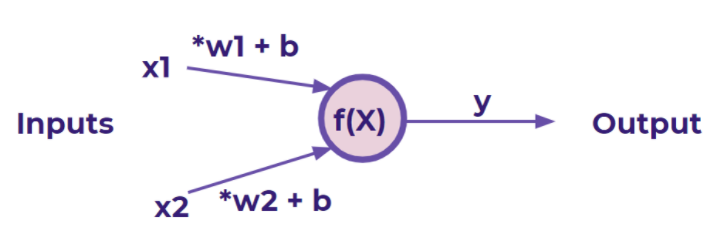
* **Input Layer [Giriş Katmanı]**  
  The input layer contains input neurons which send information to the hidden layer. [Giriş katmanı, gizli katmana bilgi gönderen giriş nöronlarını içerir.]
* **Hidden Layer [Gizli Katman]**  
  The hidden layer is used to send data to the output layer. [Gizli katman, çıktı katmanına veri göndermek için kullanılır.]
* **Output Layer [Çıktı Katmanı]**  
  The data is made available at the output layer. [Veriler çıktı katmanında kullanıma sunulur.]

- Interview Q&A

## **Activation Functions**

### **Activation Functions**

**1. [1.] What is Activation Function [Aktivasyon Fonksiyonu Nedir?]**  
You remember the picture below from the previous lessons. [Aşağıdaki resmi önceki derslerden hatırlarsınız.] The picture highlights that each neuron is characterized by its weight, bias, and activation function. [Resim, her bir nöronun ağırlığı, yanlılığı ve aktivasyon fonksiyonu ile karakterize edildiğini vurgulamaktadır.]



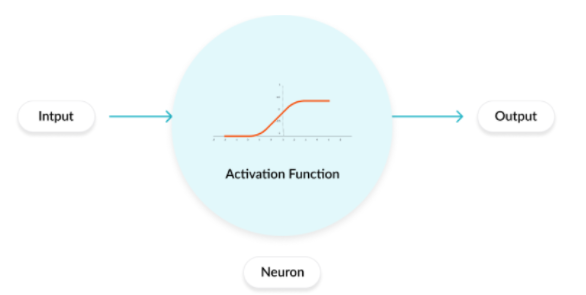
In a neural network, the input is fed to the input layer, the neurons perform a linear transformation on the input using the weights and biases. [Bir sinir ağında, giriş, giriş katmanına beslenir, nöronlar, ağırlıkları ve önyargıları kullanarak giriş üzerinde doğrusal bir dönüşüm gerçekleştirir.]

(weight \* input) + bias

Then, an activation function is applied to the above result. [Ardından, yukarıdaki sonuca bir aktivasyon işlevi uygulanır.]

y= **activation**((weight\*input)+bias)

After that, the output from the activation function moves to the next hidden layer and the same process is repeated. [Daha sonra aktivasyon fonksiyonunun çıktısı bir sonraki gizli katmana taşınır ve aynı işlem tekrarlanır.]

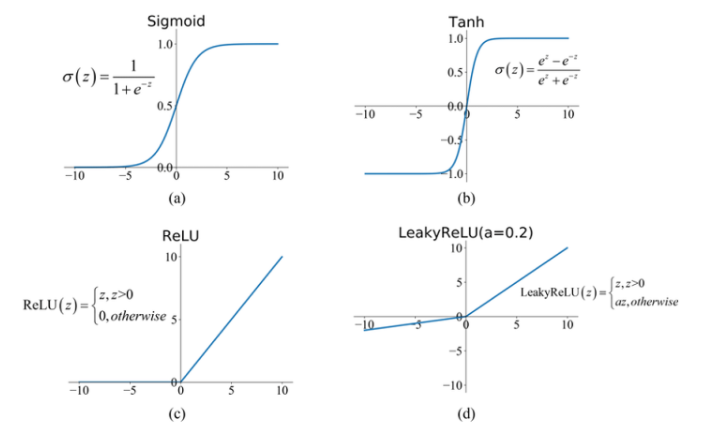


**2. [2.] Types of Activation Functions [Aktivasyon Fonksiyonlarının Türleri]**  
There are several types of activation functions. [Birkaç tür etkinleştirme işlevi vardır.] You need to choose the appropriate activation function while creating a deep learning model. [Derin öğrenme modeli oluştururken uygun aktivasyon fonksiyonunu seçmeniz gerekiyor.] The most common types are given below. [En yaygın türler aşağıda verilmiştir.]

* Sigmoid, [Sigmoid,]
* TanH/Hyperbolic Tangent, [TanH/Hiperbolik Tanjant,]
* ReLU (Rectified Linear Unit), [ReLU (Rektifiye Lineer Birim),]
* LeakyRELU. [SızdıranRELU.]

**💡Tips: [💡İpuçları:]**

* All these activation functions are **non-linear**. [Tüm bu aktivasyon fonksiyonları doğrusal değildir.]   Modern neural network models use non-linear activation functions. [Modern sinir ağı modelleri, doğrusal olmayan aktivasyon işlevlerini kullanır.]



# **Why Non-linear Activation Functions (C1W3L07)**

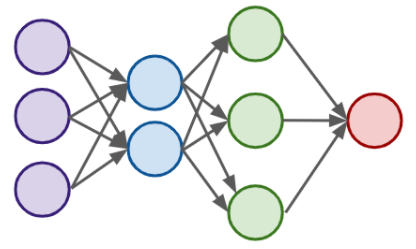
<https://www.youtube.com/watch?v=NkOv_k7r6no>

Q: What is the use of the Activation function? [S: Aktivasyon fonksiyonunun kullanımı nedir?]  
A: The activation function is used to introduce nonlinearity into the neural network so that it can learn more complex function. [C: Aktivasyon işlevi, daha karmaşık işlevleri öğrenebilmesi için sinir ağına doğrusal olmayanlığı tanıtmak için kullanılır.] Without the Activation function, the neural network would be only able to learn function, which is a linear combination of its input data. [Aktivasyon işlevi olmadan, sinir ağı yalnızca giriş verilerinin doğrusal bir birleşimi olan işlevi öğrenebilecektir.] Activation function translates the inputs into outputs. [Aktivasyon fonksiyonu, girdileri çıktılara çevirir.] The activation function is responsible for deciding whether a neuron should be activated or not. [Aktivasyon fonksiyonu, bir nöronun aktive edilip edilmeyeceğine karar vermekten sorumludur.] It makes the decision by calculating the weighted sum and further adding bias with it. [Kararı, ağırlıklı toplamı hesaplayarak ve buna yanlılığı ekleyerek verir.] The basic purpose of the activation function is to introduce non-linearity into the output of a neuron. [Aktivasyon fonksiyonunun temel amacı, bir nöronun çıktısına lineer olmayanlık getirmektir.]

- Interview Q&A

### **Multi-Class Classification [Çok Sınıflı Sınıflandırma]**

So far, you have learned what the activation function is and the types of activation functions. [Şimdiye kadar etkinleştirme işlevinin ne olduğunu ve etkinleştirme işlevlerinin türlerini öğrendiniz.]   In this lesson, you will learn how to choose the appropriate activation function for multi-class classification problems. [Bu derste, çok sınıflı sınıflandırma problemleri için uygun aktivasyon fonksiyonunun nasıl seçileceğini öğreneceksiniz.] But first, remember what is the binary classification and the multi-class classification. [Ama önce, ikili sınıflandırmanın ve çok sınıflı sınıflandırmanın ne olduğunu hatırlayın.]  
  
**1. [1.] Binary Classification [İkili Sınıflandırma]**



In a binary classification problem, we observe a single node at the output layer. [İkili sınıflandırma probleminde, çıktı katmanında tek bir düğüm gözlemleriz.]   This single node can output a binary classification between one of two possible choices, such as whether: [Bu tek düğüm, aşağıdakiler gibi iki olası seçenekten biri arasında bir ikili sınıflandırma çıktısı verebilir:]

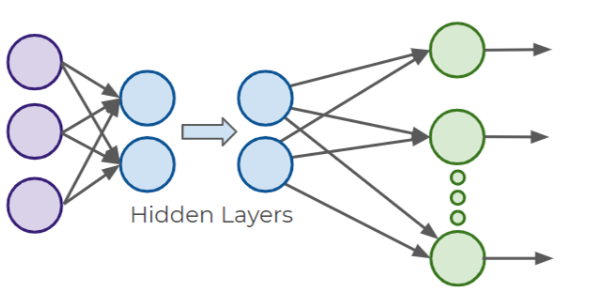
* A given email is spam or not spam. [Belirli bir e-posta spam'dir veya spam değildir.]
* A given tumor is malignant or benign. [Belirli bir tümör kötü huylu veya iyi huyludur.]

**2. [2.] Multi-Class Classification [Çok Sınıflı Sınıflandırma]**

In multi-class classification, there are classes more than two classes. [Çok sınıflı sınıflandırmada ikiden fazla sınıf vardır.]   For example: [Örneğin:]

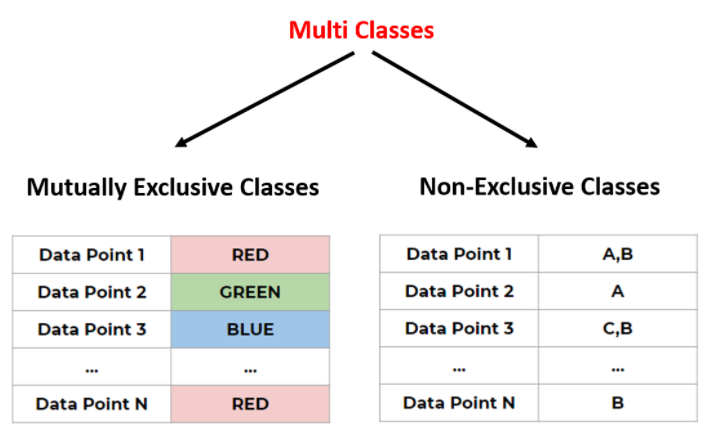
* Is this picture a dog, a cat, a horse, or a donkey? [Bu resim köpek mi, kedi mi, at mı, eşek mi?]
* Is this flower a rose, a daisy, an acacia, or a Lilac? [Bu çiçek gül mü, papatya mı, akasya mı yoksa Leylak mı?]
* Is that plane a Boeing 777, Airbus 320, Boeing 747, or Embraer 190? [Bu uçak Boeing 777, Airbus 320, Boeing 747 veya Embraer 190 mı?]

Thus, there are multiple nodes in the output layer. [Bu nedenle, çıktı katmanında birden fazla düğüm vardır.]



There are two types of multi classes: [İki tür çoklu sınıf vardır:]

* Mutually Exclusive Classes [Karşılıklı Özel Sınıflar]
* Non-Exclusive Classes [Münhasır Olmayan Sınıflar]



In Mutually Exclusive Classes each data point has a single class. [Karşılıklı Özel Sınıflarda her veri noktasının tek bir sınıfı vardır.] It can be either green, red, or blue. [Yeşil, kırmızı veya mavi olabilir.] But not blue and red at the same time. [Ama aynı anda hem mavi hem de kırmızı değil.] However, in Non-Exclusive Classes, each data point might have multiple classes. [Ancak, Özel Olmayan Sınıflarda, her veri noktasının birden fazla sınıfı olabilir.] For example data point 1 has class "A" and "B" and data point 3 has class "C" and "B". [Örneğin, veri noktası 1 'A' ve 'B' sınıfına sahiptir ve veri noktası 3 'C' ve 'B' sınıfına sahiptir.]  
  
In this lesson you will learn why you should use: [Bu derste neden kullanmanız gerektiğini öğreneceksiniz:]  
  
**Softmax** function for Mutually Exclusive Classes, [Karşılıklı Özel Sınıflar için Softmax işlevi,]  
**Sigmoid** function for Non-Exclusive Classes. [Özel Olmayan Sınıflar için Sigmoid işlevi.]

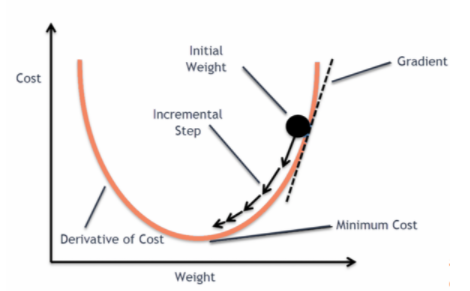
## **Cost Function, Gradient Descent and Backpropagation**

### **Cost Function and Gradient Descent [Maliyet Fonksiyonu ve Gradyan İnişi]**

In this lesson, you will learn the following concepts in Deep Learning. [Bu derste, Derin Öğrenmede aşağıdaki kavramları öğreneceksiniz.]

* Cost (Loss) Function [Maliyet (Zarar) Fonksiyonu]
* Gradient Descent [Dereceli alçalma]
* Learning Rate (Next Page) [Öğrenme Hızı (Sonraki Sayfa)]
* Adam Optimizer (Next Page) [Adam Optimize Edici (Sonraki Sayfa)]

**1. [1.] Cost Function [Maliyet fonksiyonu]**  
  
In Deep Learning, the cost function (also referred to as loss function) is used to predict how bad the model is performing. [Derin Öğrenmede, modelin ne kadar kötü performans gösterdiğini tahmin etmek için maliyet işlevi (kayıp işlevi olarak da adlandırılır) kullanılır.] In other words, a cost function is a measure of how wrong the model is in terms of its ability to predict the result. [Başka bir deyişle, maliyet fonksiyonu, modelin sonucu tahmin etme yeteneği açısından ne kadar yanlış olduğunun bir ölçüsüdür.] This is basically explained as a difference between the predicted value and the actual value. [Bu temelde tahmin edilen değer ile gerçek değer arasındaki fark olarak açıklanır.]    
  
Thus, the objective of a Deep Learning model is to find the right weights and biases that minimizes the cost function. [Bu nedenle, Derin Öğrenme modelinin amacı, maliyet fonksiyonunu en aza indiren doğru ağırlıkları ve önyargıları bulmaktır.] In other words, models learn by minimizing the cost function. [Diğer bir deyişle, modeller maliyet fonksiyonunu minimize ederek öğrenirler.]    
  
**2. [2.] Gradient Descent [Dereceli alçalma]**  
Gradient descent is an optimization algorithm used to minimize the cost function. [Gradyan iniş, maliyet fonksiyonunu en aza indirmek için kullanılan bir optimizasyon algoritmasıdır.]   It is best used when the parameters cannot be calculated analytically and must be searched for by an optimization algorithm. [Parametreler analitik olarak hesaplanamadığında ve bir optimizasyon algoritması tarafından aranması gerektiğinde en iyi şekilde kullanılır.]  
  
The name of this concept gives an idea about the meaning of it. [Bu kavramın adı, anlamı hakkında fikir vermektedir.] According to Cambridge Dictionary "Gradient" means *"how steep a slope is"* and the "Descent" means *"a movement down". [Cambridge Sözlüğü'ne göre 'Gradient', 'bir eğimin ne kadar dik olduğu' anlamına gelir ve 'Descent', 'aşağı doğru bir hareket' anlamına gelir.]*  A simple demonstration of Gradient Descent is given in the picture below. [Gradient Descent'in basit bir gösterimi aşağıdaki resimde verilmiştir.] You see the black point with the initial weight. [İlk ağırlıkla siyah noktayı görüyorsunuz.] Draw an imaginary line to the cost axis from the black point. [Siyah noktadan maliyet eksenine hayali bir çizgi çizin.] As the black point goes down while the model continues to learn, the corresponding cost value will decrease. [Model öğrenmeye devam ederken siyah nokta azaldıkça, karşılık gelen maliyet değeri düşecektir.] The final object is to reach the minimum cost value as shown in the picture. [Nihai amaç, resimde gösterildiği gibi minimum maliyet değerine ulaşmaktır.]



# **Gradient descent, how neural networks learn | Chapter 2, Deep learning**

<https://www.youtube.com/watch?v=IHZwWFHWa-w&t=3s>

Q: Explain the gradient descent. [S: Gradyan inişini açıklayın.]  
A: It is an optimization algorithm that is used to minimize some function by repeatedly moving in the direction of steepest descent as specified by the negative of the gradient is known as gradient descent. [C: Gradyanın negatifiyle belirtilen en dik iniş yönünde art arda hareket ederek bazı fonksiyonları en aza indirmek için kullanılan bir optimizasyon algoritmasıdır, gradyan inişi olarak bilinir.] It's an iteration algorithm, in every iteration algorithm, we compute the gradient of a cost function, concerning each parameter and update the parameter of the function. [Bu bir yineleme algoritmasıdır, her yineleme algoritmasında, her parametreyle ilgili olarak bir maliyet işlevinin gradyanını hesaplar ve işlevin parametresini güncelleriz.] In machine learning, it is used to update the parameters of our model. [Makine öğrenmesinde, modelimizin parametrelerini güncellemek için kullanılır.] Parameters represent the coefficients in linear regression and weights in neural networks. [Parametreler, doğrusal regresyondaki katsayıları ve sinir ağlarındaki ağırlıkları temsil eder.]

- Interview Q&A

Q: What is the cost function? [S: Maliyet fonksiyonu nedir?]  
A: A cost function describes us how well the neural network is performing with respect to its given training sample and the expected output. [C: Maliyet işlevi, verilen eğitim örneğine ve beklenen çıktıya göre sinir ağının ne kadar iyi performans gösterdiğini bize açıklar.] It may depend on variables such as weights and biases. [Ağırlıklar ve önyargılar gibi değişkenlere bağlı olabilir.] It provides the performance of a neural network as a whole. [Bir sinir ağının performansını bir bütün olarak sağlar.] In deep learning, our priority is to minimize the cost function. [Derin öğrenmede önceliğimiz maliyet fonksiyonunu minimize etmektir.] That's why we prefer to use the concept of gradient descent. [Bu yüzden gradyan iniş kavramını kullanmayı tercih ediyoruz.]

- Interview Q&A

### **Learning Rate and Adam Optimizer [Öğrenme Hızı ve Adam Optimizer]**

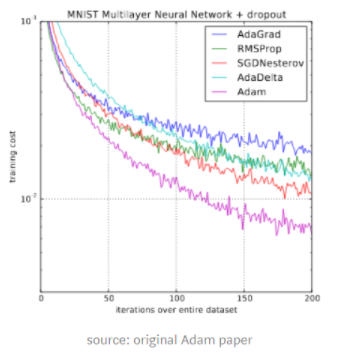
**3. [3.] Learning Rate [Öğrenme oranı]**  
  
Imagine you are walking in the top mountains and you are able to take huge steps. [En yüksek dağlarda yürüdüğünüzü ve büyük adımlar atabildiğinizi hayal edin.] Your goal is to reach the river on the ground. [Amacınız yerdeki nehre ulaşmak.]   The cost function (the error) is maximum at the top of the mountain and it is minimum at the river level. [Maliyet fonksiyonu (hata) dağın tepesinde maksimum, nehir seviyesinde minimumdur.]



The **size of your steps** can be considered as the **learning rate**. [Adımlarınızın boyutu öğrenme oranı olarak kabul edilebilir.] If the steps are large enough you reach the minimum cost function very quickly, but if it is too large you might miss the minimum level. [Adımlar yeterince büyükse minimum maliyet fonksiyonuna çok hızlı ulaşırsınız, ancak çok büyükse minimum seviyeyi kaçırabilirsiniz.] The learning rate is demonstrated as the "*incremental steps*" in the picture under the Gradient Descent title. [Öğrenme oranı, Gradient Descent başlığı altındaki resimde 'artımlı adımlar' olarak gösterilmektedir.]

**4. [4.] Adam Optimizer [Adam Doktoru]**

Adam is a method for optimization. [Adam optimizasyon için bir yöntemdir.] In other words, it is one of the way to implement the Gradient Descent algorithm. [Başka bir deyişle, Gradient Descent algoritmasını uygulama yollarından biridir.] Because there are several optimization methods to implement the Gradient Descent algorithm. [Çünkü Gradient Descent algoritmasını uygulamak için birkaç optimizasyon yöntemi vardır.] However, it is among the latest trends and perhaps the most efficient method. [Ancak son trendler arasında ve belki de en verimli yöntem.]  The picture below demonstrates the comparison of Adam with other optimization methods. [Aşağıdaki resim, Adam'ın diğer optimizasyon yöntemleriyle karşılaştırmasını göstermektedir.]



# **Learning Rate in a Neural Network explained**

<https://www.youtube.com/watch?v=jWT-AX9677k>

### **Backpropagation [Geri yayılım]**

The last theory topic you will learn during the Deep Learning course is Backpropagation. [Derin Öğrenme kursunda öğreneceğiniz son teori konusu Geri Yayılımdır.] Maybe it is the hardest one. [Belki de en zorudur.] Because the calculus and the notation of the Backpropagation are very difficult to understand. [Çünkü Geri Yayılımın hesabı ve gösterimini anlamak çok zordur.] If you are not able to understand the calculus and the notation, just try to understand the logic behind this concept. [Kalkülüs ve gösterimi anlayamıyorsanız, bu kavramın arkasındaki mantığı anlamaya çalışın.]  
  
So far you learned that the objective of a Deep Learning model is to find the right weights and biases that minimize the cost function (error). [Şimdiye kadar bir Derin Öğrenme modelinin amacının, maliyet fonksiyonunu (hatayı) en aza indiren doğru ağırlıkları ve önyargıları bulmak olduğunu öğrendiniz.] Because the model learns by minimizing the error. [Çünkü model hatayı en aza indirerek öğrenir.] And we mentioned Gradient Descent as an optimization algorithm used to minimize the error. [Ayrıca, hatayı en aza indirmek için kullanılan bir optimizasyon algoritması olarak Gradient Descent'den bahsettik.] In this respect, **you need to move backward through a network to update the weights and biases. [Bu bakımdan, ağırlıkları ve önyargıları güncellemek için bir ağ üzerinden geriye doğru gitmeniz gerekir.]**This whole process (going back and updating the weights and biases) is called Backpropagation. [Tüm bu sürece (geriye gitme ve ağırlıkları ve önyargıları güncelleme) Geri Yayılım denir.]

# **What is backpropagation really doing? | Chapter 3, Deep learning**

<https://www.youtube.com/watch?v=Ilg3gGewQ5U>

Q: What is Backpropagation? [S: Geri Yayılım nedir?]  
A: [A:]

Backpropagation is a training algorithm which is used for multilayer neural networks. [Geri yayılım, çok katmanlı sinir ağları için kullanılan bir eğitim algoritmasıdır.] It transfers the error information from the end of the network to all the weights inside the network. [Ağın sonundaki hata bilgisini ağ içindeki tüm ağırlıklara aktarır.] It allows the efficient computation of the gradient. [Gradyanın verimli hesaplanmasını sağlar.]

Backpropagation can be divided into the following steps: [Geri yayılım aşağıdaki adımlara ayrılabilir:]

* It can forward propagation of training data through the network to generate output. [Çıktı oluşturmak için eğitim verilerinin yayılmasını ağ üzerinden iletebilir.]
* It uses target value and output value to compute error derivative concerning output activations. [Çıkış aktivasyonlarına ilişkin hata türevini hesaplamak için hedef değeri ve çıkış değerini kullanır.]
* It can backpropagate to compute the derivative of the error concerning output activations in the previous layer and continue for all hidden layers. [Önceki katmandaki çıktı aktivasyonlarına ilişkin hatanın türevini hesaplamak için geri yayılabilir ve tüm gizli katmanlar için devam edebilir.]
* It uses the previously calculated derivatives for output and all hidden layers to calculate the error derivative concerning weights. [Çıktı için önceden hesaplanmış türevleri ve ağırlıklarla ilgili hata türevini hesaplamak için tüm gizli katmanları kullanır.]
* It updates the weights. [Ağırlıkları günceller.]

- Interview Q&A