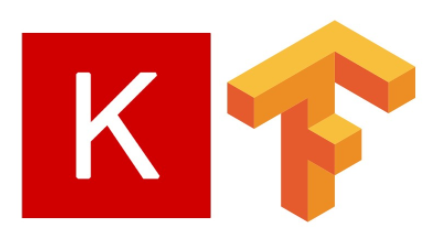
## **TF/Keras Syntax Basics**

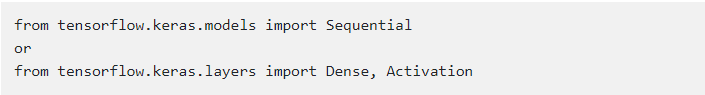
### **Keras vs TensorFlow**



Tensorflow is the most popular library used to build Deep Learning models. [Tensorflow, Derin Öğrenme modelleri oluşturmak için kullanılan en popüler kitaplıktır.] The community of TensorFlow is extremely vast and supportive, particularly because it’s an open-source platform. [TensorFlow topluluğu, özellikle açık kaynaklı bir platform olduğu için son derece geniş ve destekleyicidir.] The number of commits and forks on the GitHub repository of TensorFlow is enough to let us understand the widespread popularity of the framework. [TensorFlow'un GitHub deposundaki taahhüt ve çatalların sayısı, çerçevenin yaygın popülaritesini anlamamıza izin vermek için yeterlidir.] However, it is not that easy to work with Tensorflow. [Ancak Tensorflow ile çalışmak o kadar kolay değil.]

On the other end, Keras is a high-level API that is built on top of TensorFlow. [Öte yandan Keras, TensorFlow'un üzerine inşa edilmiş üst düzey bir API'dir.] Keras uses Tensorflow for its backend. [Keras, arka ucu için Tensorflow'u kullanır.] It is extremely user-friendly and much easier than TensorFlow. [Son derece kullanıcı dostudur ve TensorFlow'dan çok daha kolaydır.]

You will use Keras in the projects throughout this course. [Bu kurs boyunca projelerde Keras'ı kullanacaksınız.]  However you will not install Keras. [Ancak Keras'ı yüklemeyeceksiniz.]   You will install Tensorflow and reach the Keras API via Tensorflow  just like in the following examples: [Tensorflow'u yükleyecek ve aşağıdaki örneklerde olduğu gibi Tensorflow aracılığıyla Keras API'sine ulaşacaksınız:]



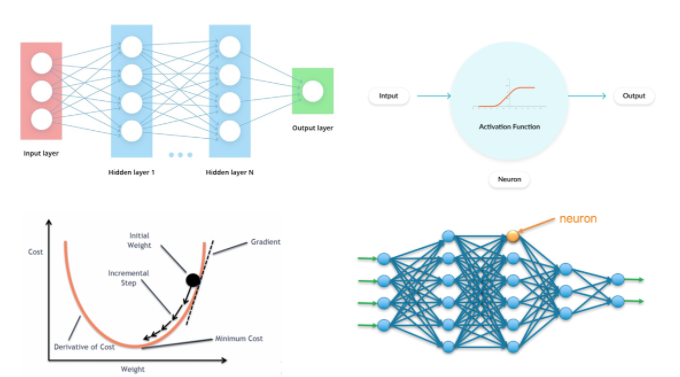
That is why we named the lessons as "*Tensorflow/Keras ...."* in the course. [Bu yüzden derste dersleri 'Tensorflow/Keras ....' olarak adlandırdık.] In this lesson, you will watch the video about TensorFlow and Keras. [Bu derste TensorFlow ve Keras ile ilgili videoyu izleyeceksiniz.] In the following lesson, you will learn how to install Tensorflow on your own computer. [Aşağıdaki derste, Tensorflow'u kendi bilgisayarınıza nasıl kuracağınızı öğreneceksiniz.]

Q: What are the deep learning frameworks or tools? [S: Derin öğrenme çerçeveleri veya araçları nelerdir?]  
A: Deep learning frameworks or tools are: [Y: Derin öğrenme çerçeveleri veya araçları şunlardır:]

Tensorflow, Keras, Chainer, Pytorch, Theano & Ecosystem, Caffe2, CNTK, DyNetGensim, DSSTNE, Gluon, Paddle, Mxnet, BigDL [Tensorflow, Keras, Chainer, Pytorch, Theano ve Ekosistem, Caffe2, CNTK, DyNetGensim, DSSTNE, Gluon, Paddle, Mxnet, BigDL]

- Interview Q&A

### **Preparing the Data**



In the previous lessons you have learned the following terms in Deep Learning: [Önceki derslerde Derin Öğrenme ile ilgili aşağıdaki terimleri öğrendiniz:]

* TensorFlow and Keras, [TensorFlow ve Keras,]
* Neuron and Neural Networks, [Nöron ve Sinir Ağları,]
* Layers in a Neural Network, [Bir Sinir Ağındaki Katmanlar,]
* Activation Functions, [Aktivasyon Fonksiyonları,]
* Cost (Loss) Function, [Maliyet (Zarar) Fonksiyonu,]
* Gradient Descent, [Dereceli alçalma,]
* Learning Rate, [Öğrenme oranı,]
* Optimization Methods (adam, rmsprop, etc) [Optimizasyon Yöntemleri (adam, rmsprop, vb.)]
* Backpropagation. [Geri yayılım.]

In this section, you will learn how to create a deep learning model using codes corresponding to these terms. [Bu bölümde, bu terimlere karşılık gelen kodları kullanarak derin öğrenme modelinin nasıl oluşturulacağını öğreneceksiniz.]  
  
You will examine this section in 3 parts. [Bu bölümü 3 kısımda inceleyeceksiniz.] The first part is about *preparing the data* you are familiar with from the ML course. [İlk bölüm, ML kursundan aşina olduğunuz verileri hazırlamakla ilgilidir.] The second part will be about *creating and training a deep learning model* where you will use the abovementioned terms. [İkinci bölüm, yukarıda belirtilen terimleri kullanacağınız bir derin öğrenme modelinin oluşturulması ve eğitilmesi ile ilgili olacaktır.] And in the third lesson, you will see the model evaluation. [Üçüncü derste ise model değerlendirmesini göreceksiniz.]

# **Getting Started with Keras**

<https://www.youtube.com/watch?v=J6Ok8p463C4>

### **Creating the Model [Modeli Oluşturma]**

In this lesson, you will start creating the model. [Bu derste, modeli oluşturmaya başlayacaksınız.] In this respect you will learn: [Bu bağlamda şunları öğreneceksiniz:]

* What is a Keras Sequential() model, [Keras Sequential() modeli nedir,]
* How to create the model and add layers to it, [Model nasıl oluşturulur ve ona katmanlar eklenir,]
* How to define the number of neurons at each layer, [Her katmandaki nöron sayısı nasıl tanımlanır,]
* How to define the activation function. [Aktivasyon fonksiyonu nasıl tanımlanır.]

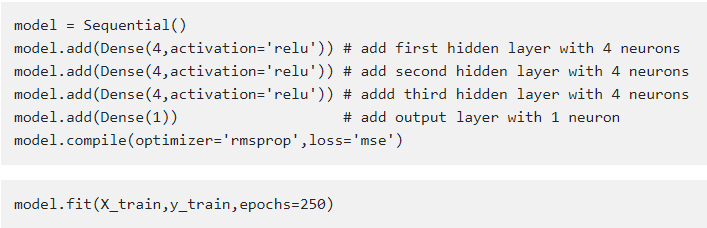
# **Create an artificial neural network with Keras**

<https://www.youtube.com/watch?v=Boo6SmgmHuM>

### **Training the Model**

# **Train an artificial neural network with Keras**

<https://www.youtube.com/watch?v=EHbay6hDZGA>



**What is Batch Size? [Parti Boyutu nedir?]**

You have already learned what epoch is. [Çağın ne olduğunu zaten öğrendin.] But this time you see another term which is called **batch size**. [Ancak bu sefer parti boyutu adı verilen başka bir terim görüyorsunuz.] Batch size is a term used in deep learning and refers to the number of training examples utilized in one iteration. [Toplu iş boyutu, derin öğrenmede kullanılan bir terimdir ve bir yinelemede kullanılan eğitim örneklerinin sayısını ifade eder.] Because you can’t pass the entire dataset into the neural net at once. [Çünkü tüm veri kümesini tek seferde sinir ağına geçiremezsiniz.] You need to divide the dataset into a number of batches or sets. [Veri kümesini birkaç toplu iş veya kümeye bölmeniz gerekir.] Don't forget: *"The smaller the batch size the longer the training takes." [Unutmayın: 'Grup boyutu ne kadar küçükse, eğitim o kadar uzun sürer.']*

**What is Epochs? [Epoch nedir?]**

According to the Cambridge Dictionary, the epoch means: *"a long period of time, especially one in which there are new developments and great change" [Cambridge Sözlüğü'ne göre, çağ şu anlama gelir: 'uzun bir zaman dilimi, özellikle de yeni gelişmelerin ve büyük değişikliklerin olduğu bir dönem']*

In Deep Learning terminology the one epoch means, the entire dataset is passed**forward** and **backward** through the neural network once. [Derin Öğrenme terminolojisinde tek dönem, tüm veri kümesinin sinir ağı üzerinden bir kez ileri ve geri geçirildiği anlamına gelir.] In other words, an epoch refers to one cycle through the full training dataset. [Başka bir deyişle, bir dönem, tam eğitim veri kümesi boyunca bir döngüyü ifade eder.] Training a neural network usually takes more than a few epochs. [Bir sinir ağını eğitmek genellikle birkaç çağdan fazla zaman alır.]  
  
**Overfitting and regularization [Aşırı takma ve düzenleme]**  
  
Overfitting(not generalizing) problem occurs for all machine learning and deep learning models if **your model is doing well on training data, but not able to perform similar on test data. [Modeliniz eğitim verilerinde iyi durumdaysa, ancak test verilerinde benzer performans gösteremiyorsa, tüm makine öğrenimi ve derin öğrenme modelleri için fazla sığdırma (genelleme yapmama) sorunu oluşur.]**Strategies used to prevent overfitting is called in general as regularization methods. [Fazla takmayı önlemek için kullanılan stratejilere genel olarak düzenleme yöntemleri denir.] **Early stopping** is one of regularization techniques such that if accuracy on validation set is not improving we stop instead of going through all epochs. [Erken durdurma, doğrulama setindeki doğruluk gelişmiyorsa, tüm çağlardan geçmek yerine duracak şekilde düzenleme tekniklerinden biridir.] Also we can deactivate random nodes in network and make a single model behave like different network everytime during training. [Ayrıca ağdaki rastgele düğümleri devre dışı bırakabilir ve eğitim sırasında her zaman tek bir modelin farklı ağ gibi davranmasını sağlayabiliriz.] This technique is called **dropout. [Bu tekniğe bırakma denir.]**

### **Regression/Classification Examples**

# **Get started with using TensorFlow to solve for regression problems (Coding TensorFlow)**

<https://www.youtube.com/watch?v=-vHQub0NXI4>

# **GPU bench-marking with image classification | Deep Learning Tutorial 17 (Tensorflow2.0, Python)**

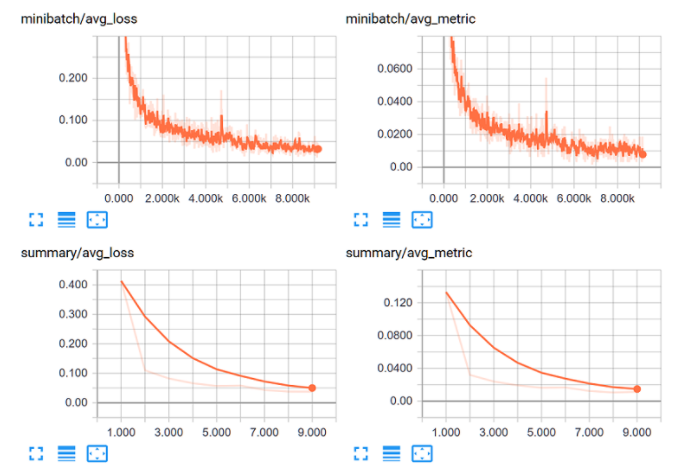
<https://www.youtube.com/watch?v=YmDaqXMIoeY&t=198s>

## **TensorBoard (Optional)**

Tensorboard is an interface used to visualize and optimize your model. [Tensorboard, modelinizi görselleştirmek ve optimize etmek için kullanılan bir arayüzdür.] Because visualizing the graph of your network is very straight-forward in TensorBoard. [Çünkü TensorBoard'da ağınızın grafiğini görselleştirmek çok basittir.]    
  
TensorBoard provides the following requirements: [TensorBoard aşağıdaki gereksinimleri sağlar:]

* Tracking and visualizing metrics such as loss and accuracy [Kayıp ve doğruluk gibi metrikleri izleme ve görselleştirme]
* Visualizing the model graph (ops and layers) [Model grafiğini görselleştirme (operasyonlar ve katmanlar)]
* Viewing histograms of weights, biases, or other tensors as they change over time [Ağırlıkların, yanlılıkların veya diğer tensörlerin histogramlarını zamanla değiştikçe görüntüleme]
* Projecting embeddings to a lower-dimensional space [Gömmeleri daha düşük boyutlu bir alana yansıtma]
* Displaying images, text, and audio data [Görüntüleri, metinleri ve ses verilerini görüntüleme]
* Profiling TensorFlow programs [TensorFlow programlarının profilini oluşturma]
* And much more [Ve daha fazlası]

(source for the above list: <https://www.tensorflow.org/tensorboard>)



# **Tensorboard Introduction | Deep Learning Tutorial 16 (Tensorflow2.0, Keras & Python)**

<https://www.youtube.com/watch?v=9SdLOcGnebU&t=1s>