deep learningin amacı bize en az hatayı verecek ağırlık ve bias değerlerini bulmak.

başlangıçta ağırlık değerleri random olarak atanıyor. sonra hata değerini düzeltmek maksadıyla bunlarda değişikliğe gidiliyor

ml de sonraki öncekinden öğrenen yöntem olan boosting gibi. her layer bir öncekinden öğrenerek gidiyor

aktivasyon fonkyonlarının yeri nöronların içi..

bir önceki nörondan bilgiler geldi--> ağırlıkla çarpıldı -->  bias eklendi--> aktivasyon fonksiyonundan beslendi--> ve aktivasyon fonksiyonu neyse ondan çıktı aldı.

aktivasyon fonksiyonu: nörona gelen bilgiyi amaca bağlı olarak sınırlandırmak için veya veri geçsin veya geçmesin diye. (nörona input olarak sıfır değer gelmesi o nöronun aktive olmaması demektir.) nöronun aktive olup olmaması, olacaksa hangi değerle aktif olmasını belirliyor.

Backpropagation işlevi türev alarak yapılan bir işlev olduğu için, linear aktivasyon fonk. backpropagation kullanılmaz.girişi lineer olan aradaki tüm katmanları lineer olan ve sonucu lineer olan bir modelin lineer regresyondan hiç bir farkı yoktur. (uç uca eklenmiş lineer regresyonların birlikte hareket etmesi gibi bir şey olur)o yüzden   linear aktivasyon fonk., artık hidden layer larda kullanılmıyor.linear aktivasyon output layer'da (en son çıktı alınan yerde) kullanılıyor.

**sigmoid aktivasyon fonk.** gelen değerlerin hepsini 0 ile 1 arasına topluyor. ağırlıklar başta random atandığı için birden atlamaların önüne geçmek maksadıyla değerleri 0-1 arasına sıkıştırıyor.

**sigmoid aktivasyonun en önemli dez avantajı:** tüm verileri 0 ile 1 arasına sıkıştırdığı için öğrenme çok yavaş oluyor.

**Relu Activation function:** sıfırına altındaki değerler 0'a çevriliyor. Bunun dezavantajı, eksi değerler 0'a dönüşmesi o nöronu öldüreceği için bir çoğu bu şekilde olursa öğrenme gerçekleşmeyebilir.

**Leaky Relu Activation function**: Relu acti. fonksiyonundaki ölü neron problemini ortadan kaldırıyor ama negatif inputlar için tutarlı predictionlar sunamıyor.

multi claficasyon problemlerinde Softmax activation kullanılıyor. ve output layer da kullanılıyor.

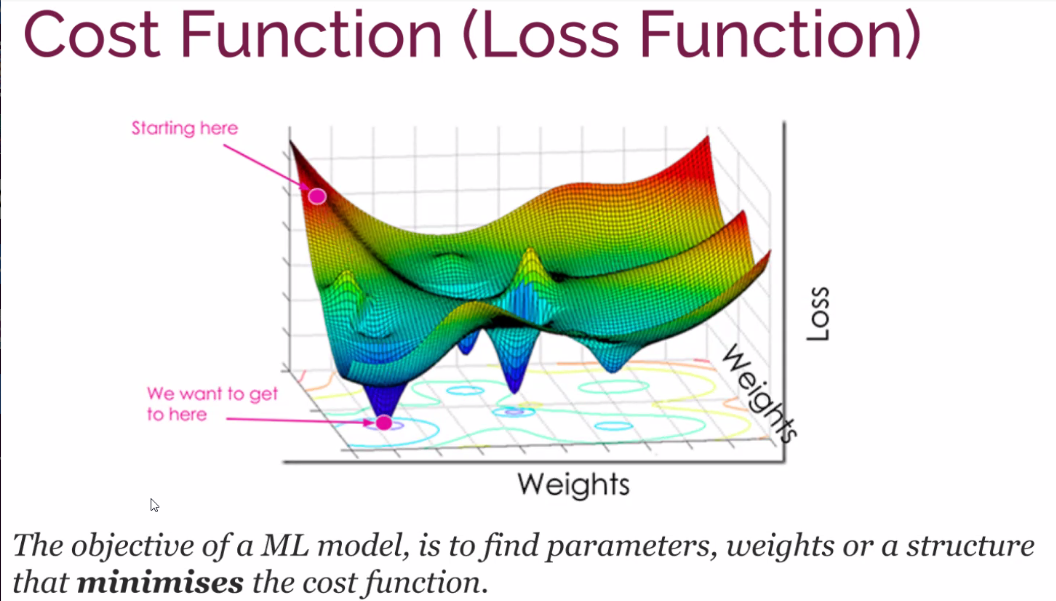
**Multclass Classification:**  
bir data sadece bir class'a ait ise buna **mutual exclusive classes** denir.  
eğer birden fazla class'a ait olabiliyorsa buna **non-exclusice classes** denir.

**Multclass Classification:**  
bir data sadece bir class'a ait ise buna **mutual exclusive classes** denir.  
eğer birden fazla class'a ait olabiliyorsa buna **non-exclusice classes** denir.**Mutually exclusive'de;**Softmax activasyon function kullanılır. bu da bize ilgili classlara ait olma olasılıklarını verir. bu **olasılık toplamı 1'i verir**ve hangi class'ın olasılığı büyükse prediction olarak bize onu verir.**Non-exclusive'de :**olasılıkların toplamı **1 olmak zorunda değildir**.  mesela bir output A 0.6 oranla classına ve 0.8 ile B'ye ait ise o çıktı hem A hem de B'ye aittir. ama C class'ının oranı 0.1 ise C classına ait değildir deriz.

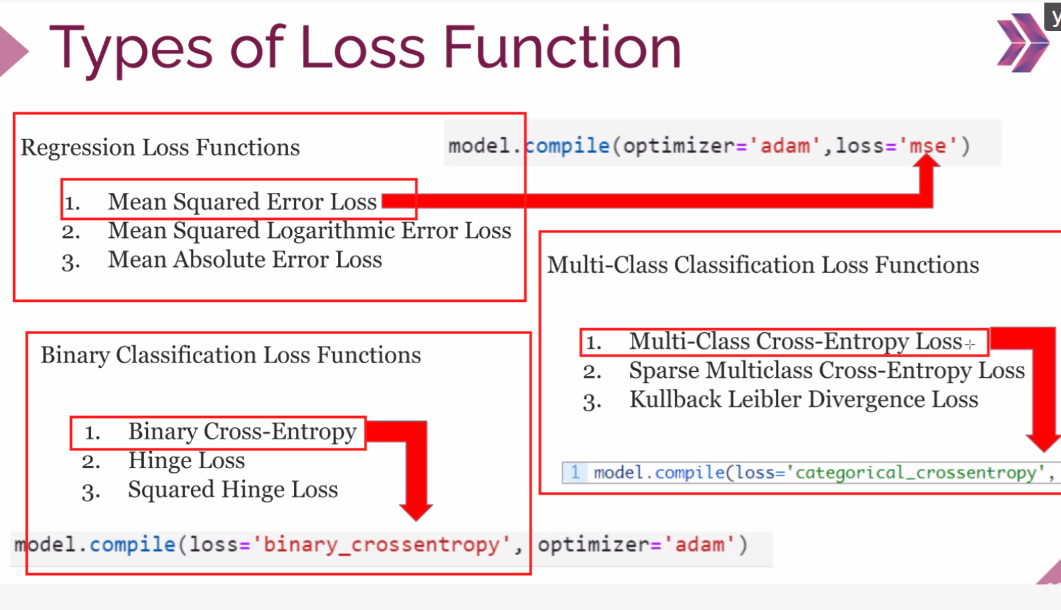
**cost function:**hatayı düzeltmek için kullanıyoruz.

**COST VE LOSS FONKSİYONLARI ARASINDAKİ FARK:**

The terms **cost**and **loss functions**almost refer to the same meaning.   
But, loss function mainly applies for a single training set as compared to the cost function which deals with a penalty for a number of training sets or the complete batch. ...   
The loss function is a **value which is calculated at every instance.**

****

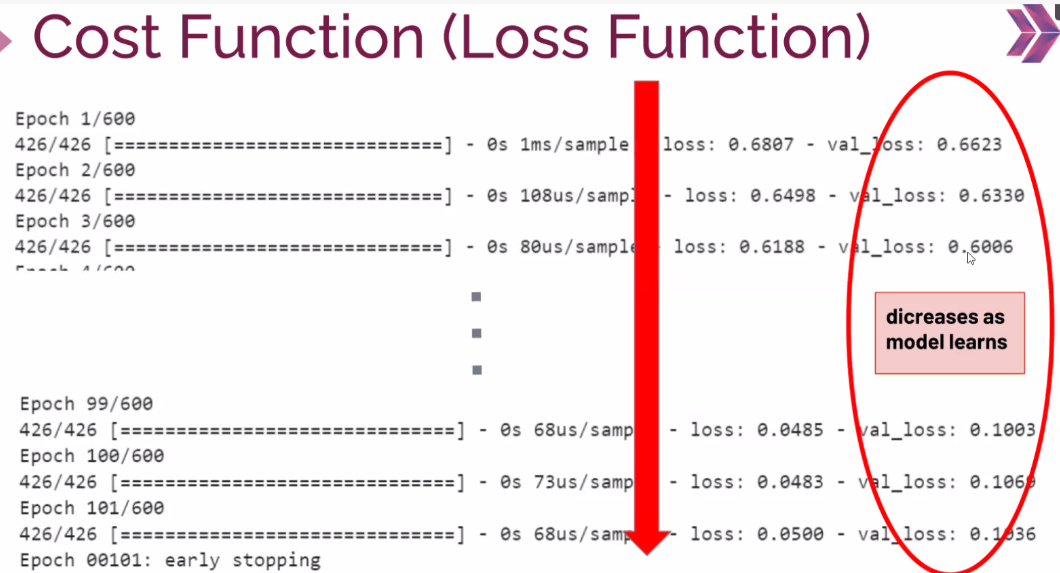
Cost function daki amacımız: hatanın minimun olduğu yerdeki bias değerini bulmaktır.



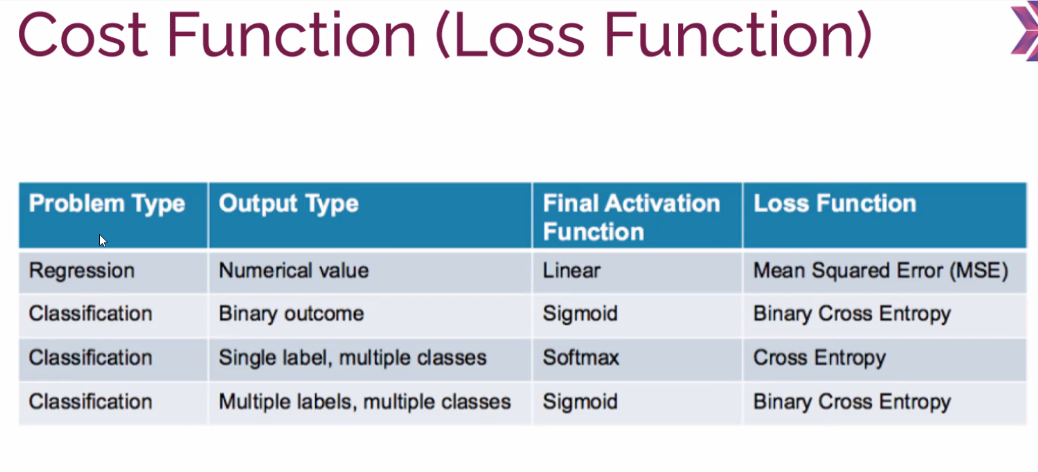
**fit'imin içinde hem eğitim yaptırıyorum hem validate (test) yaptırıyorum.epochs**= 600 --> modelin kaç kere çalıştırılacağı..

sigmoid’i hidden featurelarda kullanırsak öğrenme yavaşlar.

**epoch**'un her iterasyonunda loss function değeri düşe düşe devam ediyor.



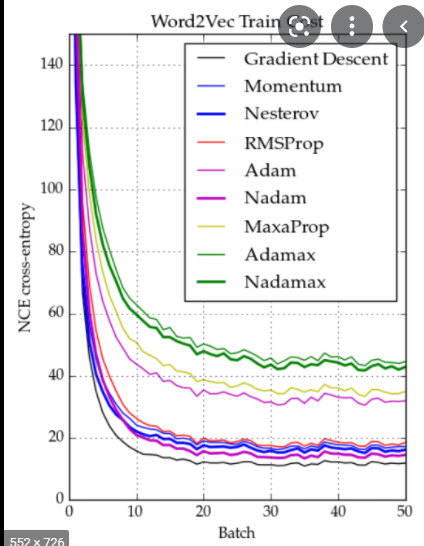
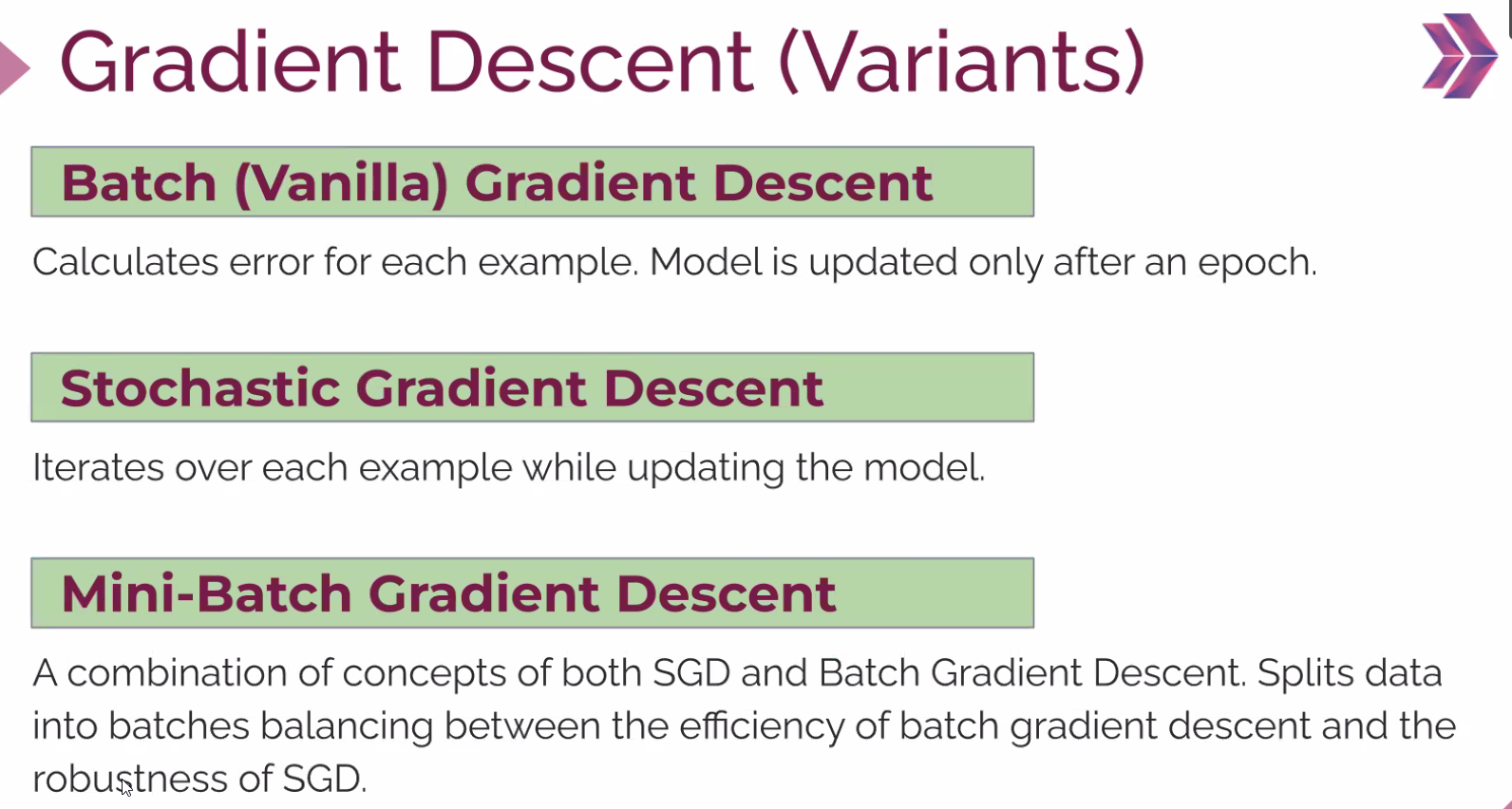
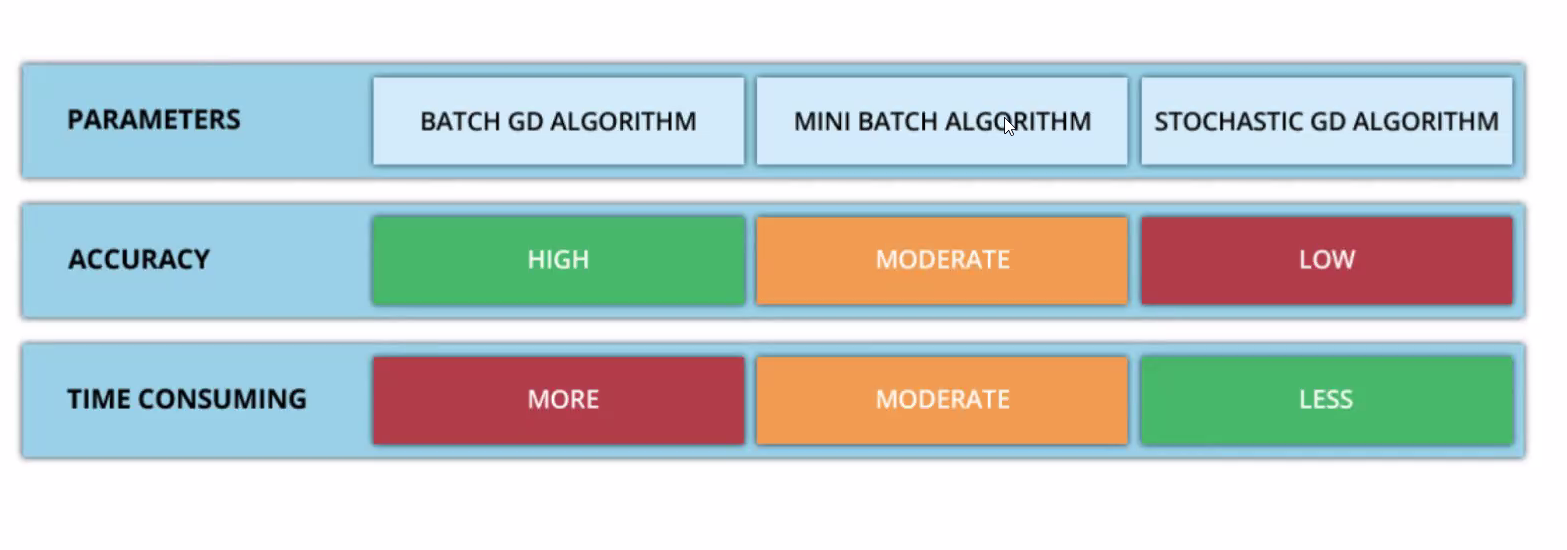
**Tüm öğrendiklerimizin özeti:**

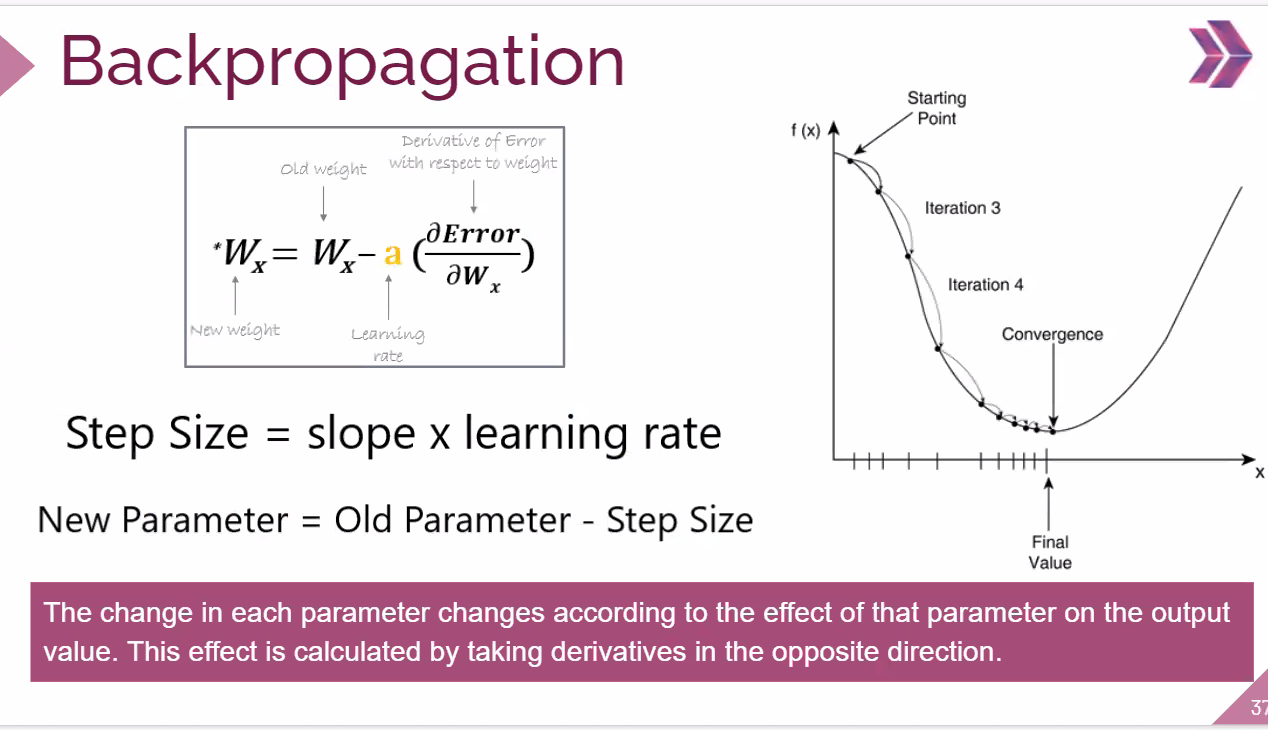


bir gözlem birden fazla class a ait olabiliyorsa sigmoid kullanılıyor. sigmoid bir gözlemi her bir class için ayrı ayrı değerlendiriyor

*https://medium.com/geekculture/different-activation-functions-for-deep-neural-networks-you-should-know-ea5e86f51e84*

Her çalıştırma için modeli kaydetmek gerekir. En düşük cost'u veren model ile devam edilir.

Genelde mini batch kullanılıyor. 



Slope bizim türevimiz.

