Eğer iki ünit birlikte çalışırsa çıkışa daha fazla tepki verir.

Bias değeri bir nöronun aktif olmadan önce ne kadar yukarı çıkmasını istiyoruz. Thereshold gibi düşünülebilir. (yanlılık fonksiyonu)

Değeri aldık ağırlıkla çarptık bias ekledik ve aktivasyon fonksiyonuna sokuyoruz.

Hidden layer :

Aktivasyon fonksiyonları : sigmoid, tanh, relu, leaky relu

Lost fonksiyonu : çapraz entropi kaybı

Mse : ortalama kare hatası :

Bce : tek çıkılşı ağ ise binary

Cce : çok çıkışlı ağ ise categorical kullanılır

Backpropagation : asıl çıktıyı ve istenen çıktıyı dikkate alarak sinir ağındaki ağırlıkları gücellemek için kullanılan bir yöntemdir. Her bir ağırlığa göre türev, zincir kuralı kullanılarak hesaplanır.

1. Adım : bir küme eğitim verisi alın ve kaybı hesaplamak için ileriye doğru ilerleyin
2. Adım : her ağırlığa göre kaybın derecesini elde etmek için kaybı tekrar geriye doğru yayın.
3. Adım : ağın ağırlıklarını güncellemek için gradyanları kullanın

Gradient descent : j(w) hata maliyeti, hatanın min olduğu nokta

Optimizasyon algoritmaları : SGD (stocastik)

Adam : momentuma göre çalışmaktadır.

Overfitting : eğer veri sizin sinir ağınızın çözebileceğinden çok çok büyükse çok hızlı bir şekilde overfitting yapacaktır. Ağı ezberlemesidir.

Dropout : iki nöron arasındaki kuvvet güçlendiğinde diğer nöronlar aktif olamıyor ve orada da bir ezberleme kavramı olmuş oluyor. Bunun için genelleştirilmesi gerekir.

Early stopping : overfitting engellenmesi için

Augmentation : veri artırma, çoğaltma

1. Adım : bir eğitim verisi kümesi alınır
2. Adım : denk gelen kaybı elde etmek için, ileri yayılım gerçekleştirilir.
3. Adım : gradyanları elde etmek için kayba geri yayılım uygulanır.
4. Adım : ağın ağırlıklaırnı güncellemek için gradyanlar kullanılır.

Anahtar kelimeler : başlangıç parametrelerinin ayarlanması, transfer öğrenme, normalizasyon, tensör ve learning rate