

5) Regularization

L2 regularization $\|w\|^2$

L1 regularization $\|w\| \rightarrow$ A lot of zero w .

Varyans yüksek ise **regularization** çözüm yollarından biridir. Loss fonksiyonuna "Penalty" termler eklenerek yapılır.

↳ Neural network'de **L2** çok daha fazla kullanılır.

* Logistic Regression $\Rightarrow \min_{w,b} j(w,b)$

$$j(w,b) = \frac{1}{m} \sum_{i=1}^m \mathcal{L}(\hat{y}^{(i)}, y^{(i)}) + \frac{\lambda}{2m} \|w\|_2^2$$

Regularization parameter.

* Neural Network'de

L2 Regularization:

$$J(\omega^{[1]}, b^{[1]}, \dots, \omega^{[L]}, b^{[L]}) = \frac{1}{m} \sum_{i=1}^m L(\hat{y}^{(i)}, y^{(i)}) + \underbrace{\frac{\lambda}{2m} \sum_{l=1}^L \|\omega^{[l]}\|_F^2}_{\text{Regul. Term}}$$

$$d\omega^{[l]} = (\text{from backprop}) + \frac{\lambda}{m} \omega^{[l]}$$

$$\omega^{[l]} = \omega^{[l]} - \alpha \left[(\text{from backprop}) + \frac{\lambda}{m} \omega^{[l]} \right]$$

$$\omega^{[l]} = \omega^{[l]} - \frac{\alpha \lambda}{m} \omega^{[l]} - \alpha (\text{from backprop})$$

! Weight matrisi çok large olduğu için cezalandırır aslında.

* Neural Network'de

Dropout Regularization: Dropout is a technique where randomly selected neurons are ignored during training. Farklı Dropout yöntemleri olsa da en common metot "Inverted Dropout" dur. Aşağıda Inverted Dropout gösterilmiştir. Bu yöntem only used training and is not used when evaluating the test set.

Regularization mantığı ne?

→ Smaller networks are trained. Bazı ünitler (nodes) eleniyor ve kullanılmıyor çünkü.

keep_prob = 0.8 olursa örneğin; 0.2 chance of eliminating any hidden unit.

İlk olarak; $d3 = \text{np.random.rand}(a3.\text{shape}[0], a3.\text{shape}[1])$ \leftarrow keep_prob bir dropout vektör oluşturulur. Daha sonra; $a3 = \text{np.multiply}(a3, d3)$ yapılır. $d3$, True=1 veya False=0 döndürür. Son olarak da $a3 /= \text{keep_prob}$ işlemleri yapılır.

Ömek

50 units \rightarrow 10 units shut off (keep prob = 0.8)
 $z^{[4]} = w^{[4]} \cdot a^{[3]} + b^{[4]}$: nöronun aktifliği

50'ten 10 unit'i eledik, bu durumda expected value of $z^{[4]}$ azalır.

$z^{[4]}$ 'ün expected value'sunu azaltmamak için şerh adım olarak $a^{[3]}$, keep prob'a bölünür. Bu da, aşağı yukarı $z^{[4]}$ 'ü aynı expected value'da tutar.

! ★ Gradient descent'in her iterasyonunda farklı hidden unit'ler zero yapılabilir.

$d3$ vektörü'nde hangi unit'in zero olacağını belirler.

! No dropout in Test set.

EK bilgi: Number of neurons in the previous layer gives us the number of columns of the weight matrix, and the number of neurons in the current layer gives us the number of rows in the weight matrix.

! Her layer için keep prob. değişebilir. Fazla unit'in olduğu layerlarda keep prob daha düşük tutulabilir.

! Output layerde keep prob = 1 olabilir yani overfit durumu olmadığı için.

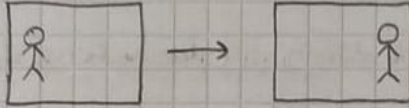
! Input layer için çok yüksek olmak istenmez. Aksi durumda feature'lerden bazıları elenebilir.

! Computer vision'da sıkça kullanılır. Çünkü, input size çok fazladır. Fakat genellikle yeterli data yoktur.

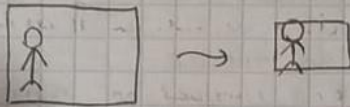
* Other Regularization

Techniques:

1) Data Augmentation: (Gogoltra)



Horizontally flipping the image.



Zoom image

Distort



Distort image.

2) Early stopping: Stopping the training of your neural network earlier.

