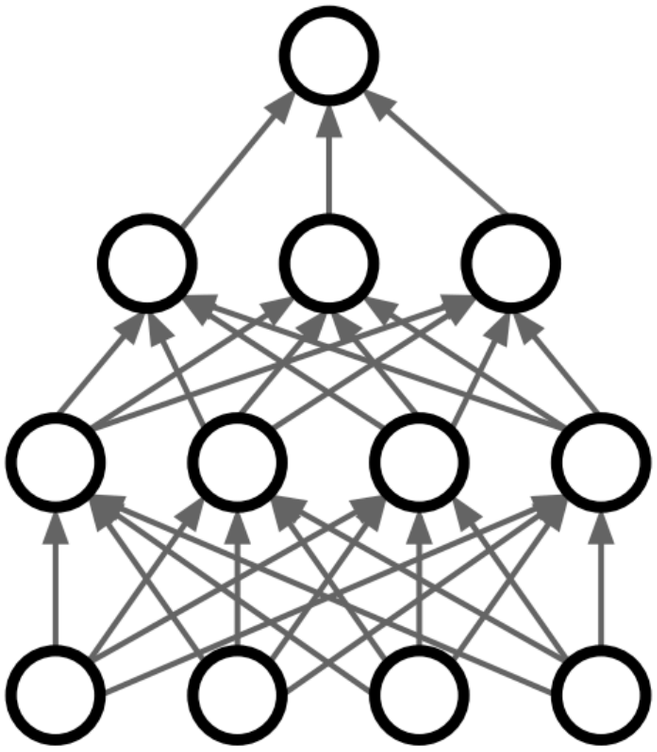
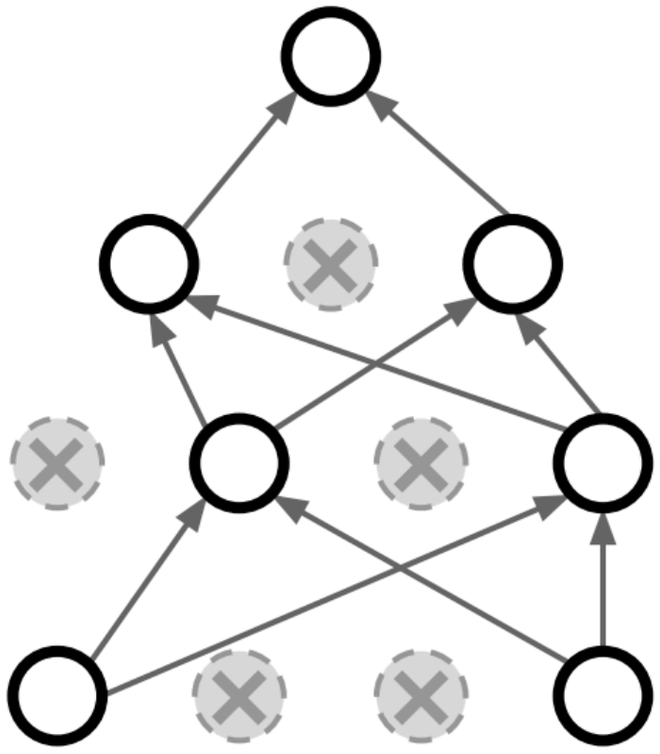
Tricks in Deep Learning

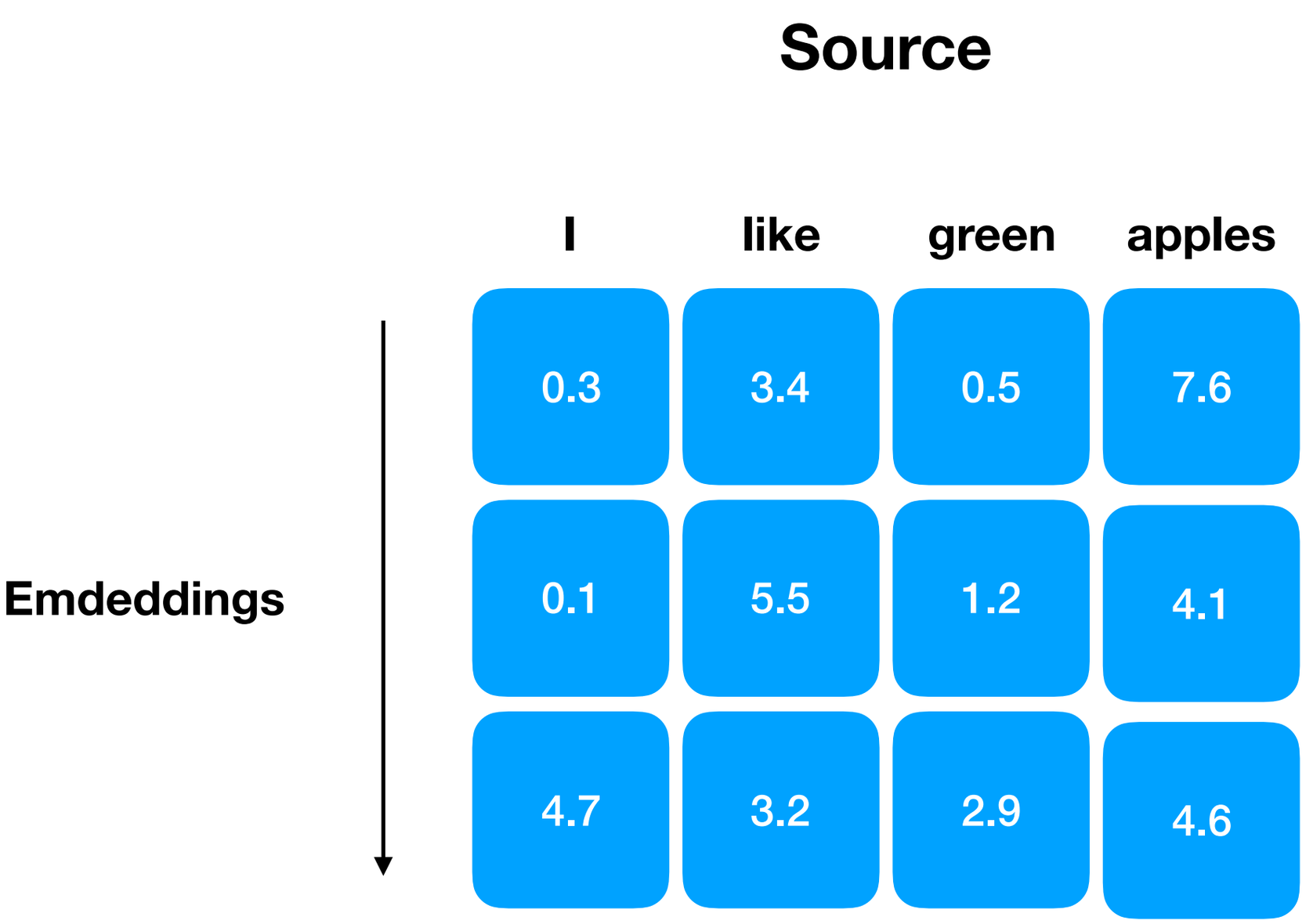
Dropout





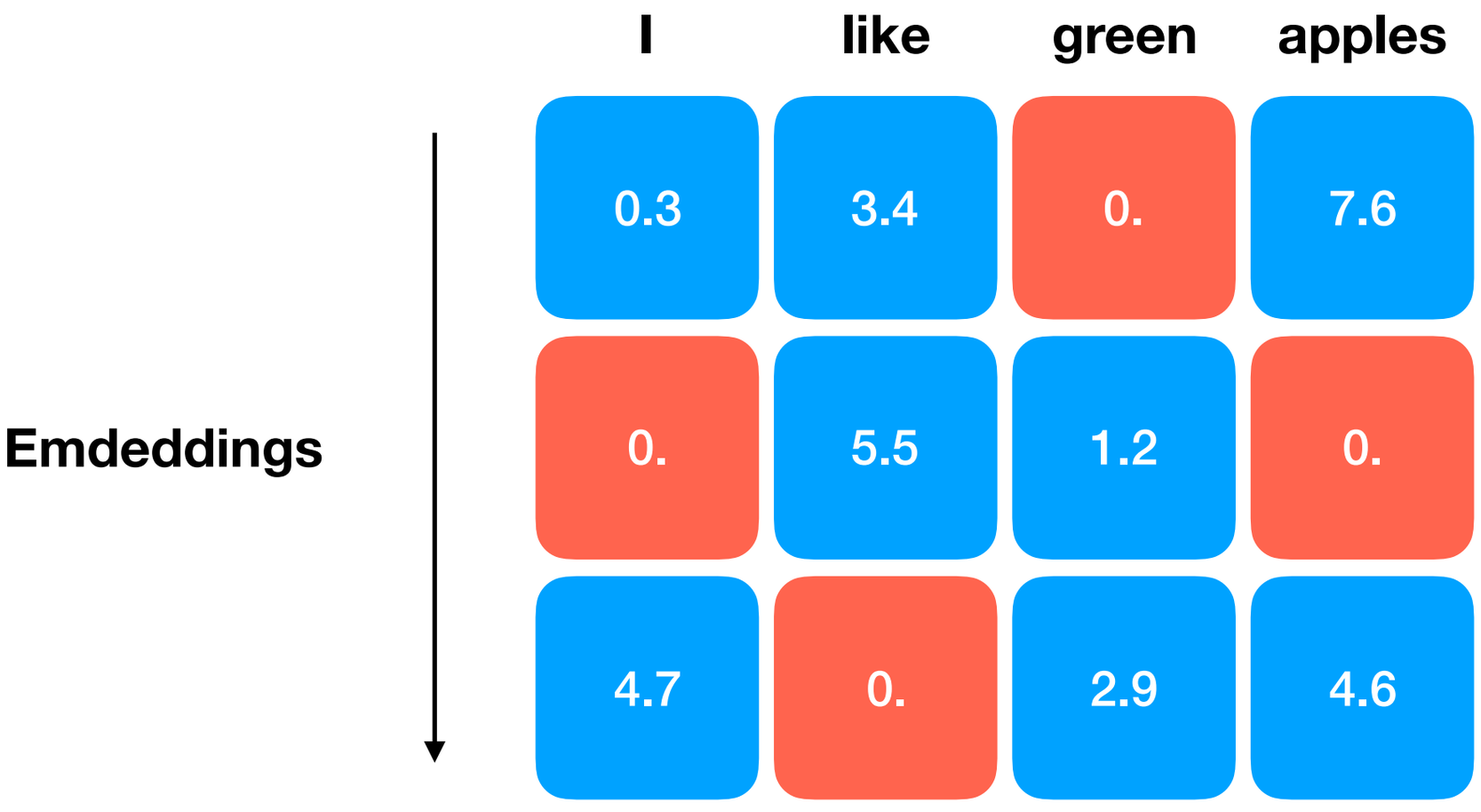

has an ear
has a tail
is furry
has claws
mischievous look
X
X
X
cat score

Dropout

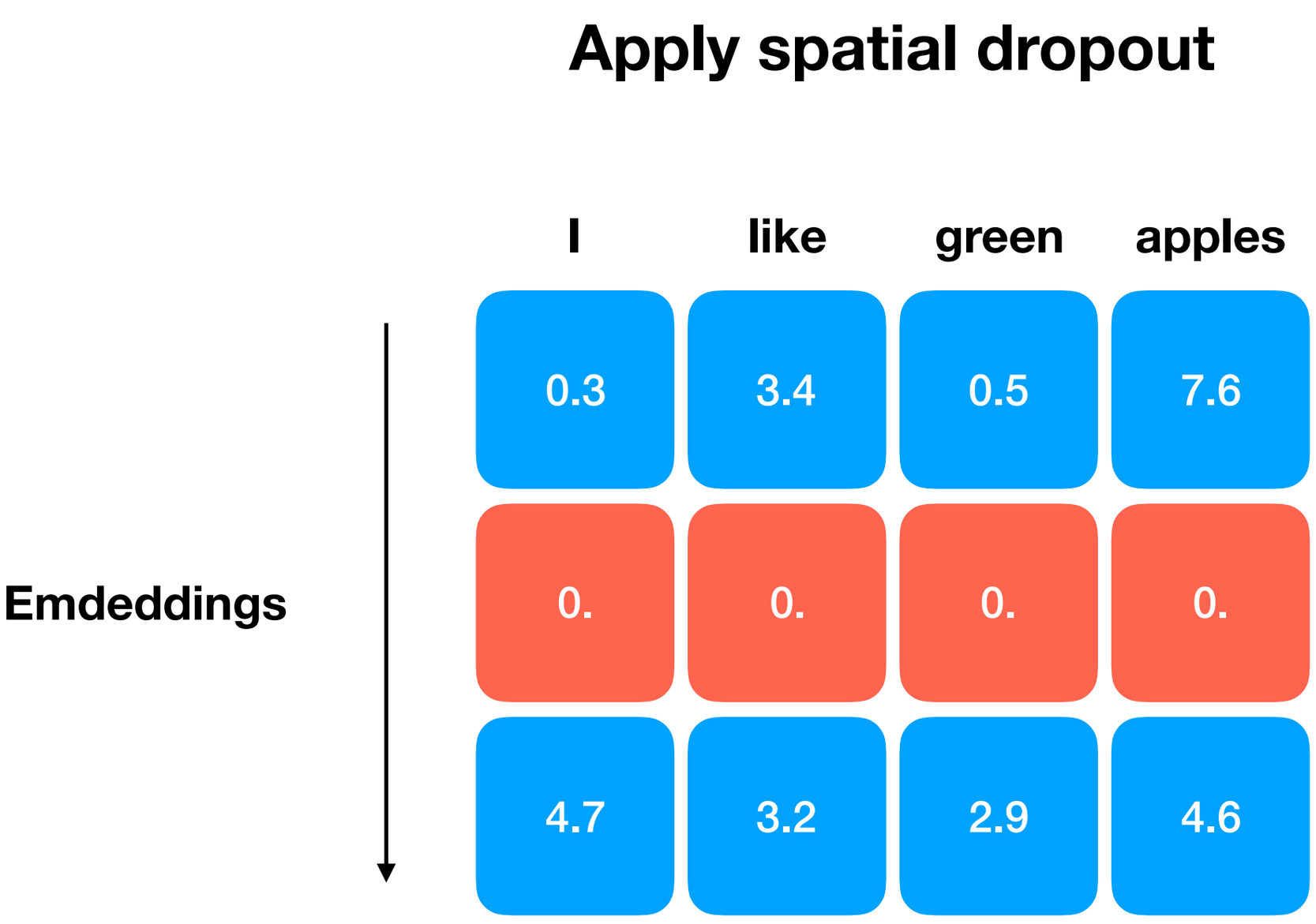

Source
I
like
green
apples
Emdeddings
0.3
3.4
0.5
7.6
0.1
5.5
1.2
4.1
4.7
3.2
2.9
4.6

Dropout

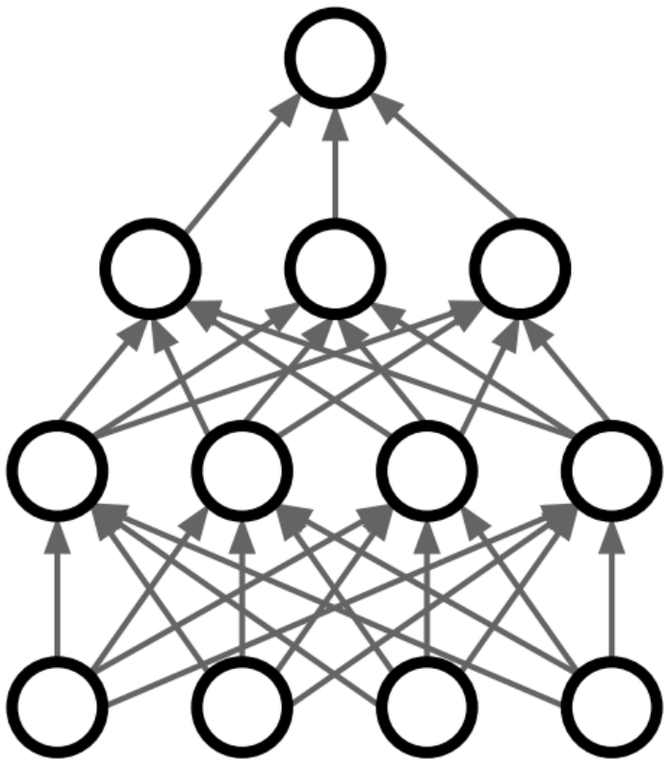
Apply dropout

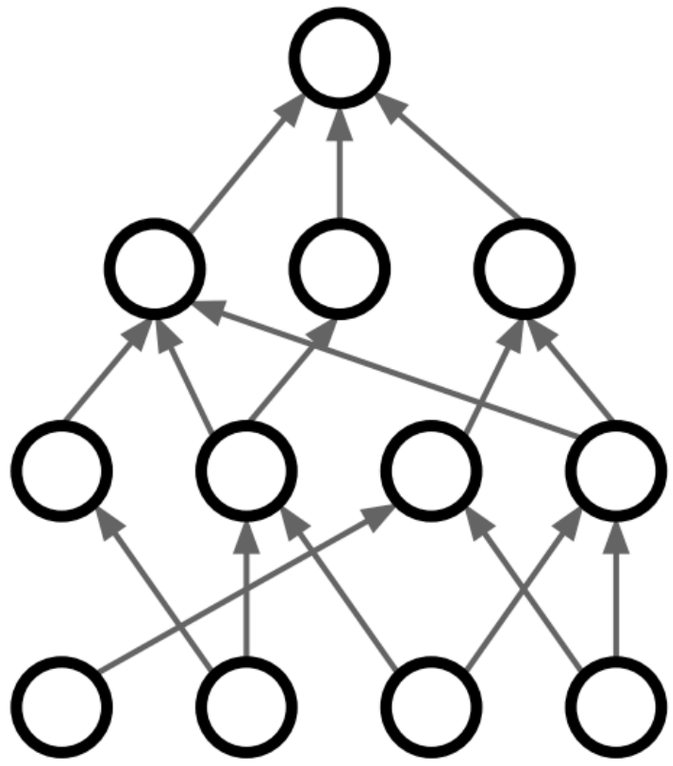


Spatial Dropout

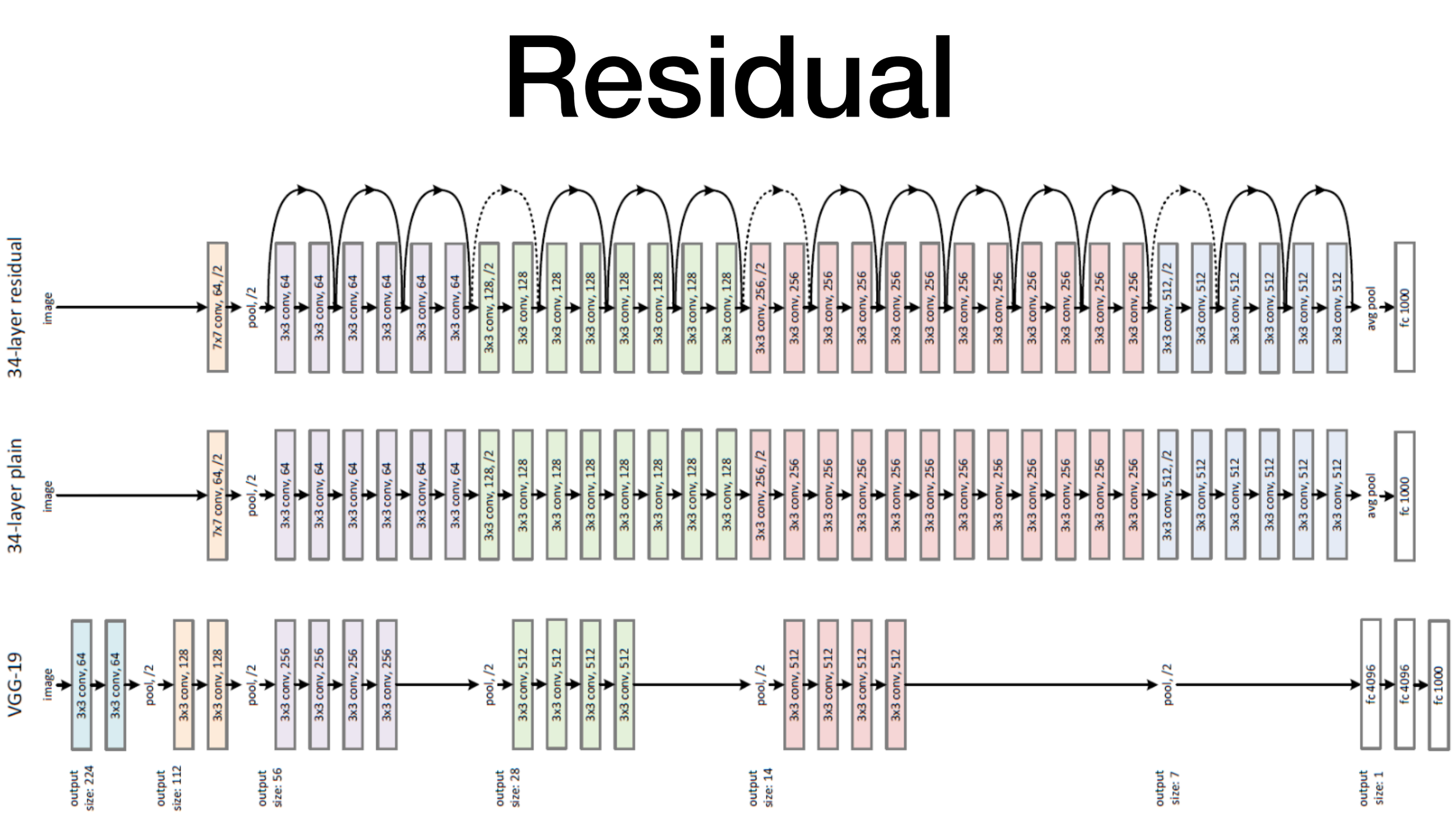


DropConnect



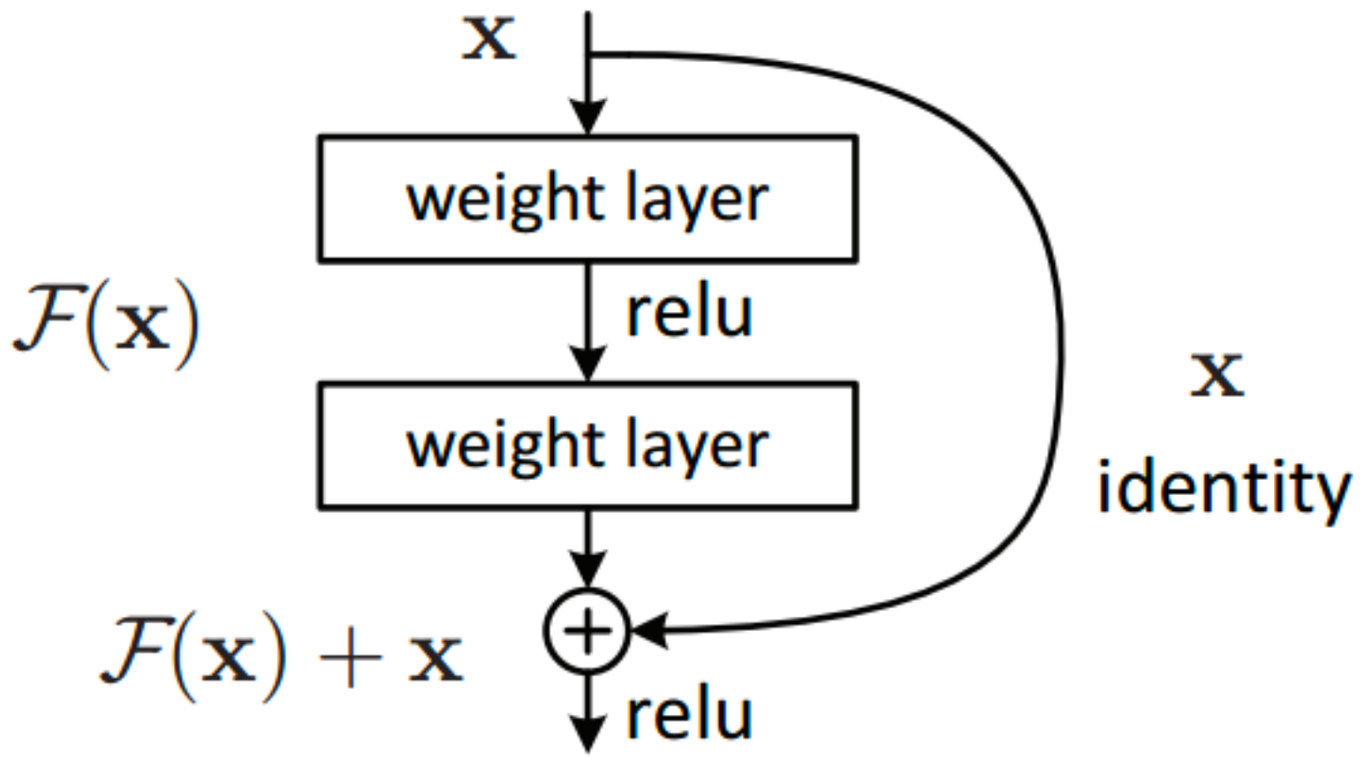


Residual

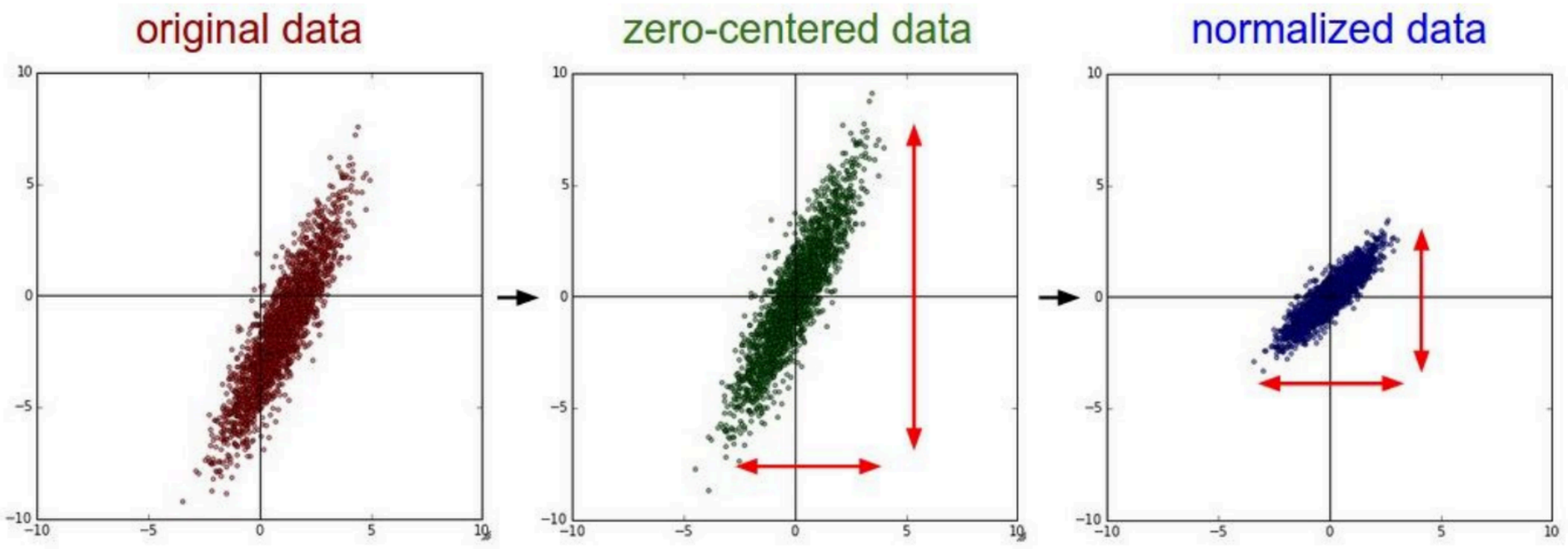


	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	**25.03**

Residual


x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

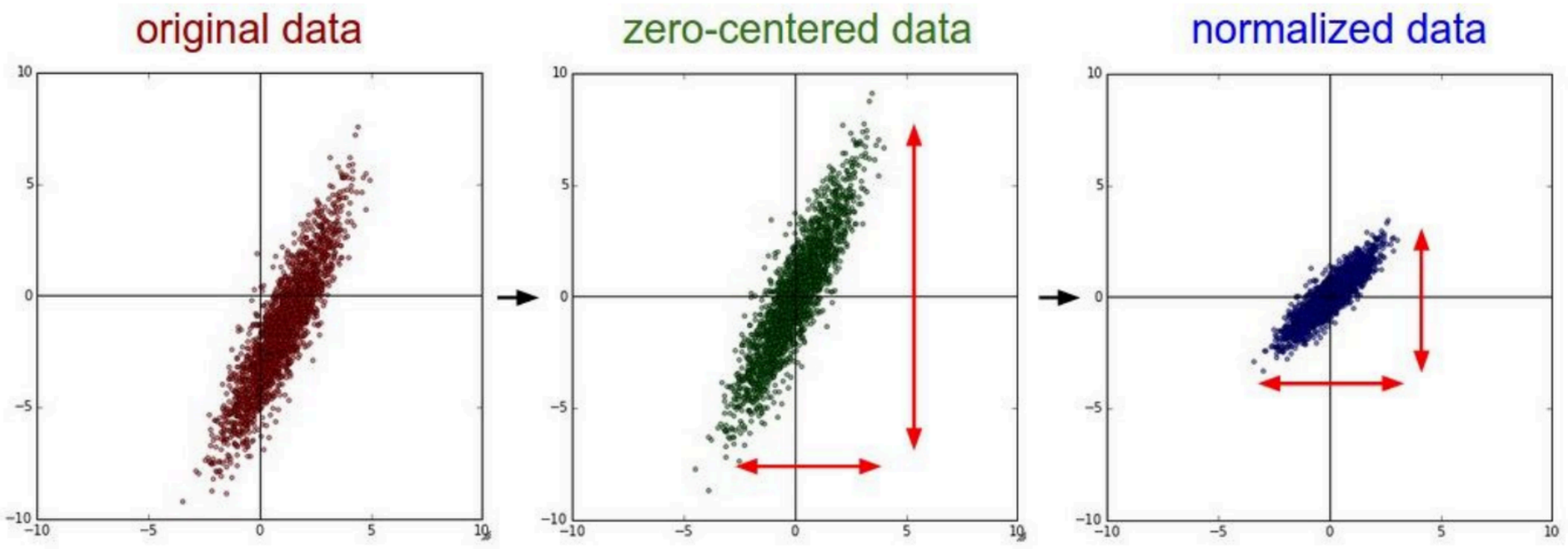
Normalization



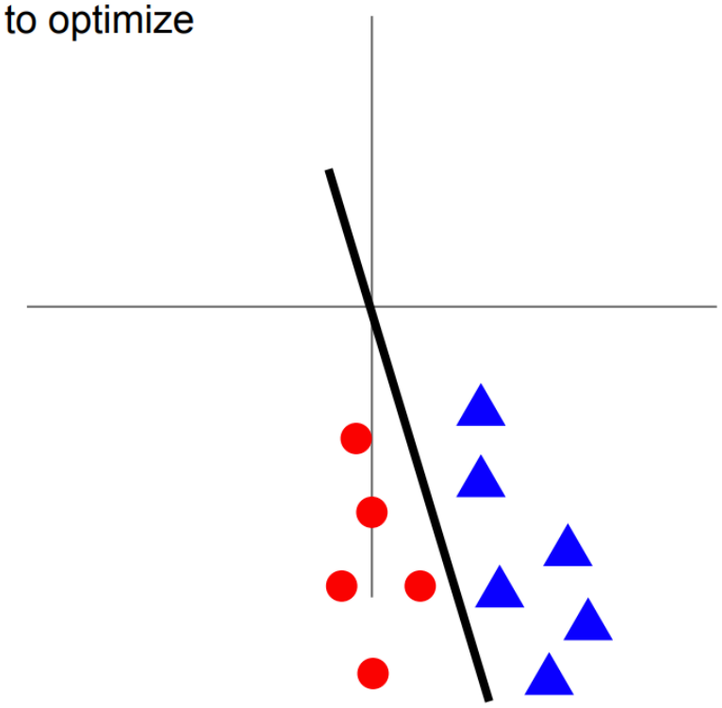

```
X -= np.mean(X, axis = 0)
```

```
X /= np.std(X, axis = 0)
```

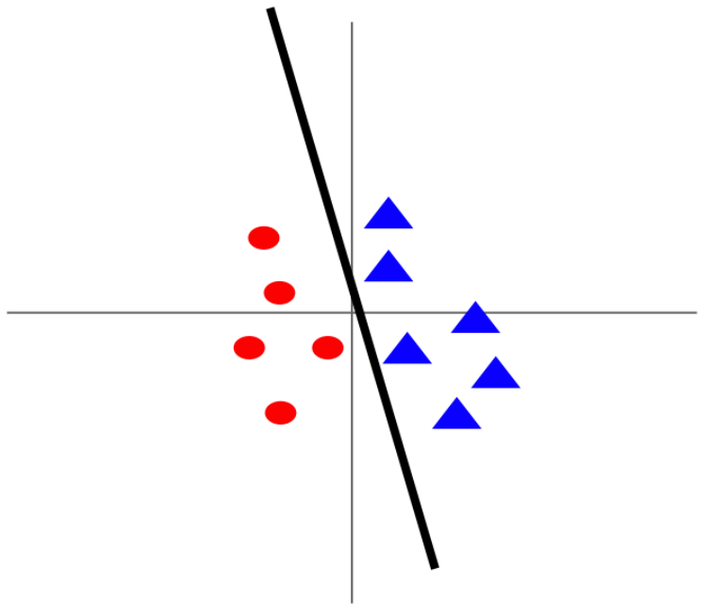
Normalization



Before normalization: classification loss very sensitive to changes in weight matrix; hard to optimize



After normalization: less sensitive to small changes in weights; easier to optimize



Batch Normalization

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Learnable

Batch Normalization

Во время предсказания батч-нормализация является линейным слоем:

$$\hat{x} = \frac{x - \mathbb{E}[x]}{\sqrt{\mathbb{D}[x] + \epsilon}}$$

$$y = \gamma \cdot \hat{x} + \beta$$

$$y = \frac{\gamma}{\sqrt{\mathbb{D}[x] + \epsilon}} \cdot x + (\beta - \frac{\gamma \mathbb{E}[x]}{\sqrt{\mathbb{D}[x] + \epsilon}})$$

$\mathbb{E}[x]$ и $\mathbb{D}[x]$ вычисляются по всему обучающему множеству. На практике статистики вычисляются во время обучения экспоненциальным средним: $E_{i+1} = (1 - \alpha)E_i + \alpha E_{\mathcal{B}}$

Training Tips

Step 1: Check initial loss
Step 2: Overfit a small sample
Step 3: Find LR that makes loss go down
Step 4: Coarse grid, train for ~1-5 epochs
Step 5: Refine grid, train longer
Step 6: Look at loss curves