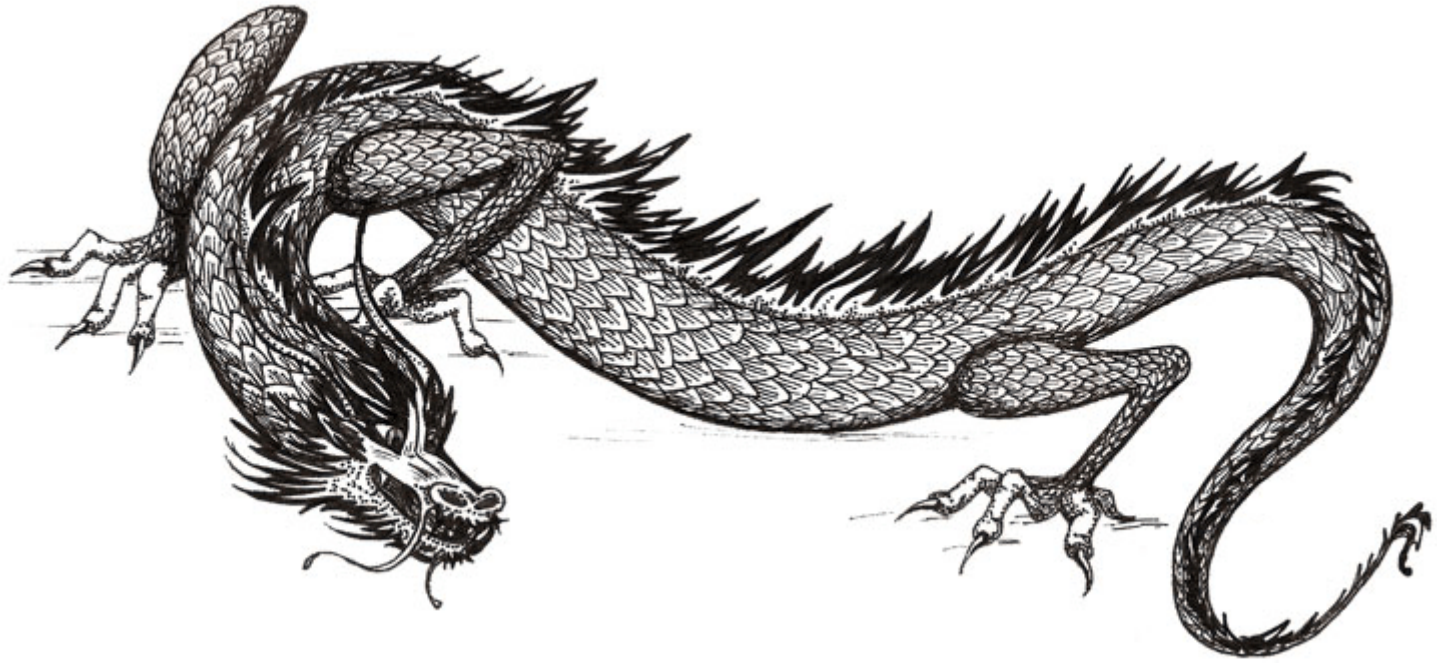


其他训练Tricks

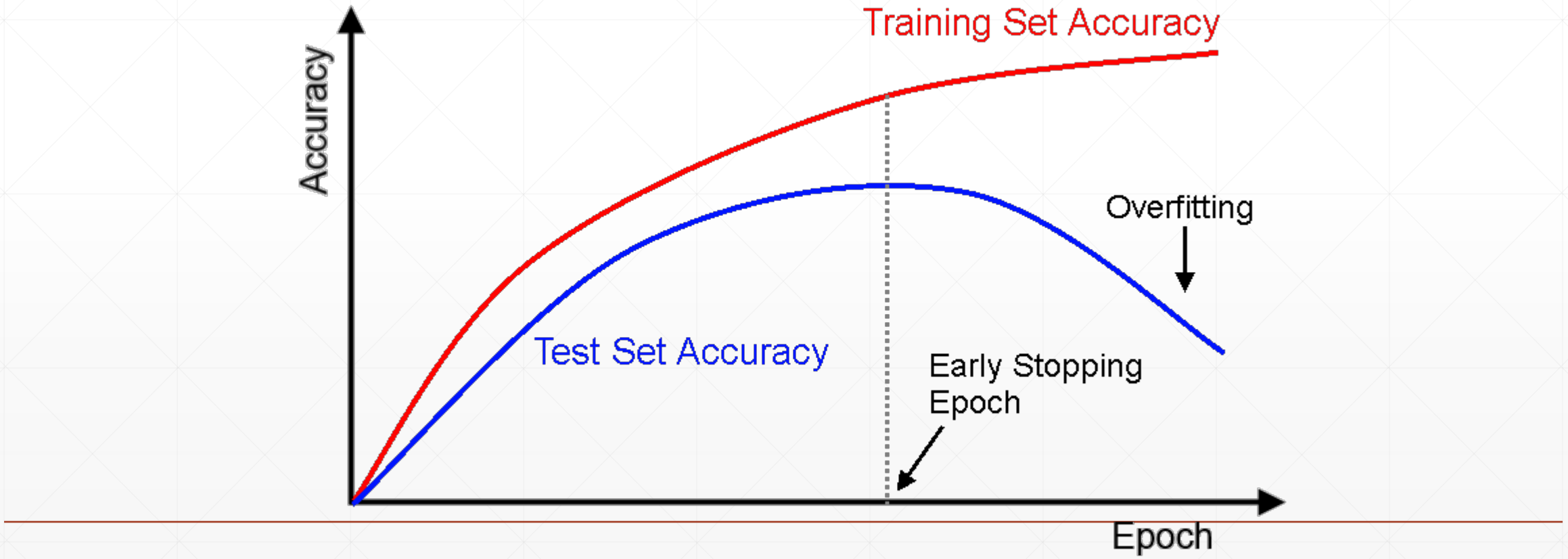
主讲：龙良曲

Tricks

- Early Stopping
- Dropout
- Stochastic Gradient Descent

Early Stopping

- Regularization

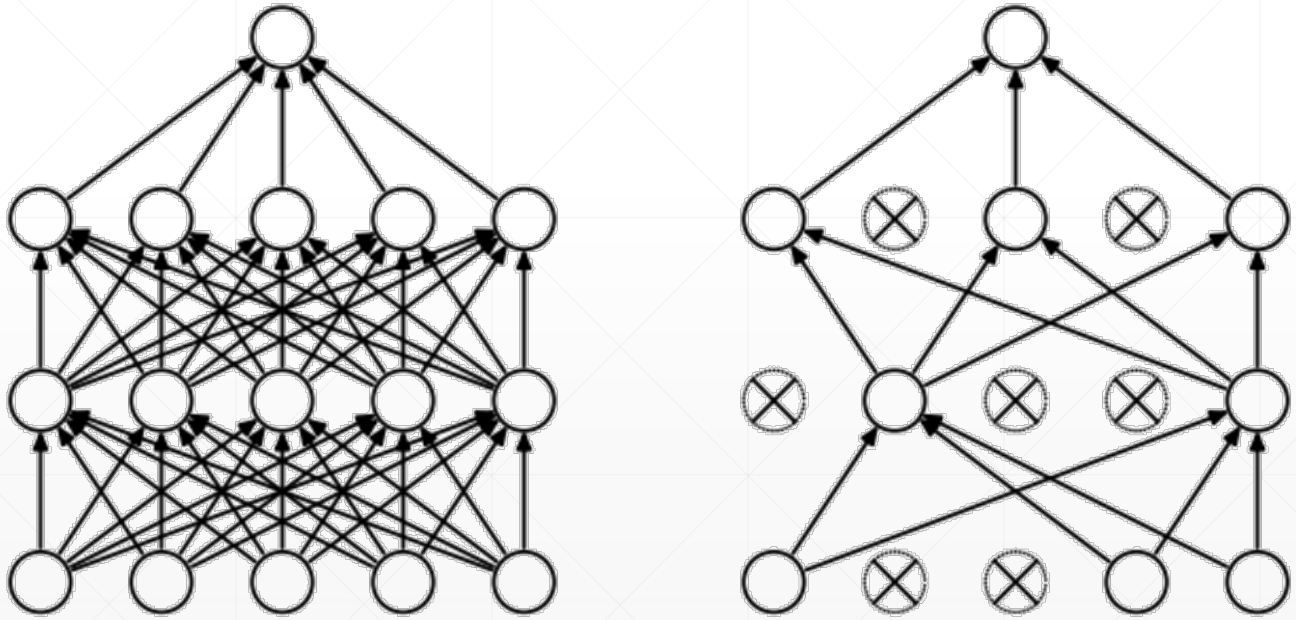


How-To

- Validation set to select parameters
- Monitor validation performance
- Stop at the highest val perf.

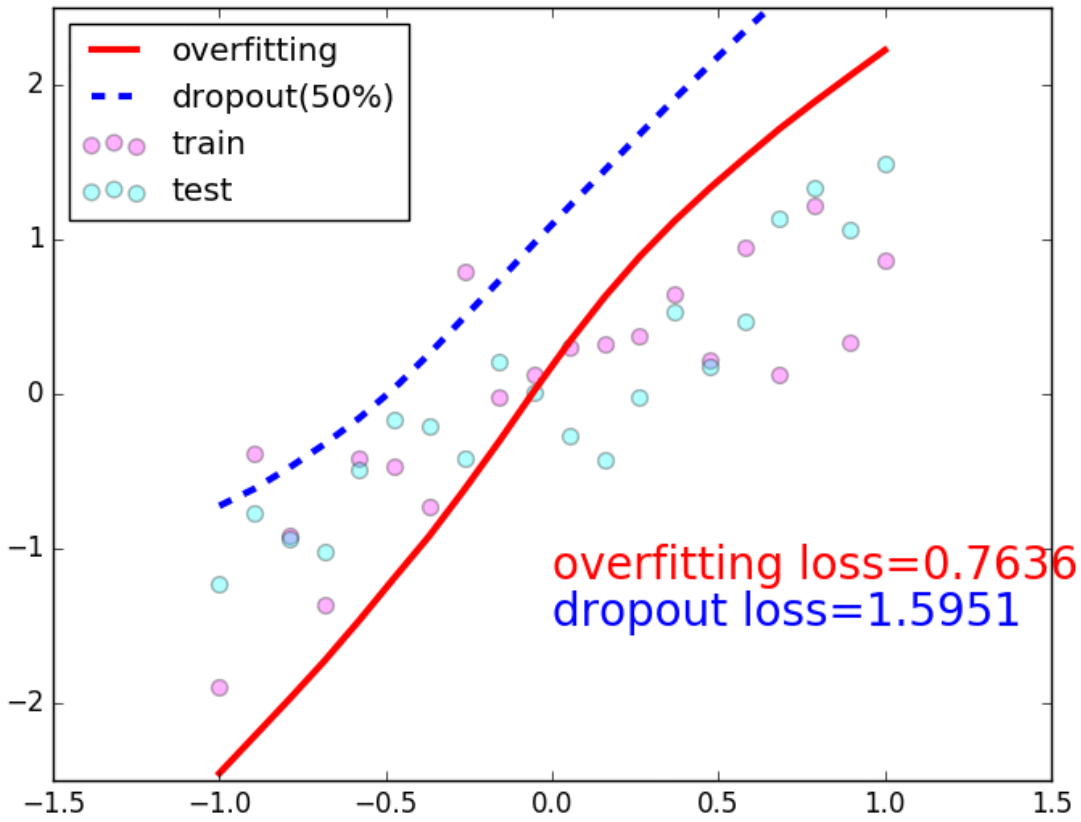
Dropout

- *Learning less to learn better*
- Each connection has $p = [0, 1]$ to lose



(a) Standard Neural Net

(b) After applying dropout.


overfitting
dropout(50%)
train
test
overfitting loss=0.7636
dropout loss=1.5951
2
1
0
−1
−2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

```
network = Sequential([layers.Dense(256, activation='relu'),
                      layers.Dropout(0.5), # 0.5 rate to drop
                      layers.Dense(128, activation='relu'),
                      layers.Dropout(0.5), # 0.5 rate to drop
                      layers.Dense(64, activation='relu'),
                      layers.Dense(32, activation='relu'),
                      layers.Dense(10)])
```

API

- *layer.Dropout(rate)*
- *tf.nn.dropout(x, rate)*

Behavior between train and test

Batch-Norm

```
for step, (x,y) in enumerate(db):

    with tf.GradientTape() as tape:
        # [b, 28, 28] => [b, 784]
        x = tf.reshape(x, (-1, 28*28))
        # [b, 784] => [b, 10]
        out = network(x, training=True)


    # test
    out = network(x, training=False)
```


JUST
DO
IT.

Stochastic Gradient Descent

- Stochastic
  - not random! 是符合一定分布的
- Deterministic

Gradient Descent

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m} \sum_{i=1}^{m} (\hat{y}^i - y^i) \cdot x_j^i$$

② Vanilla (Batch) G.D.

$$\theta_j := \theta_j - \alpha \cdot \underbrace{\frac{\partial}{\partial \theta_j} J(\theta)}_{\frac{1}{m} \sum_{i=1}^{m} (\hat{y}^i - y^i) x_j^i}$$

https://towardsdatascience.com/difference-between-batch-gradient-descent-and-stochastic-gradient-descent-1187f1291aa1

Gradient Descent

③ Stochastic G.D.

for i in range(m):

$$\theta_j := \theta_j - \alpha \cdot \underbrace{(\hat{y}^i - y^i) x_j^i}_{\text{only one example}} \frac{\nabla J}{\nabla \theta_j}$$

https://towardsdatascience.com/difference-between-batch-gradient-descent-and-stochastic-gradient-descent-1187f1291aa1

Stochastic Gradient Descent

- Not single usually
- batch = 16, 32, 64, 128…

Why 感觉应该不仅仅是因为显存不够大，因为可以每个batch前向传播完成后，保存下这个batch的loss和梯度，一个epoch后，求loss和梯度的均值，然后再更新权值。这样解决了显存不足的问题，但是当数据集很大时，参数更新会很慢。综合考量，采用SGD，并且这样做背后肯定是有数学支撑的。

¥1789.00

索泰（ZOTAC）GeForce GTX1060 X-GAMING OC吃鸡显卡/游戏电竞台式机独

6.6万+条评价　　二手有售

索泰京东自营旗舰店

自营　秒杀　券1500-200

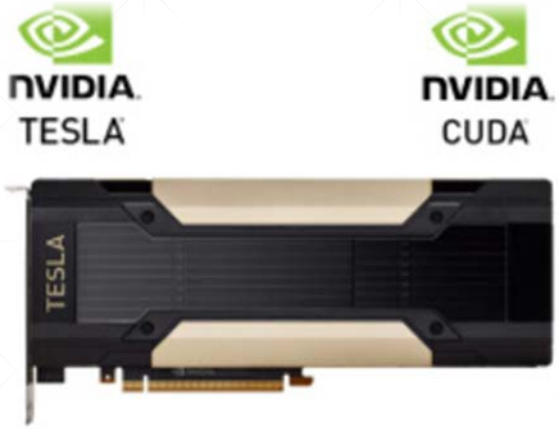
¥7599.00

索泰（ZOTAC） GTX1080Ti-11GD5X至尊PLUS OC绝地求生/吃鸡显卡

40+条评价

云轩 DIY整机专营店

满减　免邮　满199-100　险



¥80999.00

NVIDIA TESLA V100 16GB/32GB HBM2/人工智能深度学习GPU运算 16GB HBM2

0条评价

子希DIY电脑专营店

免邮

下一课时

卷积神经网络

Thank You.