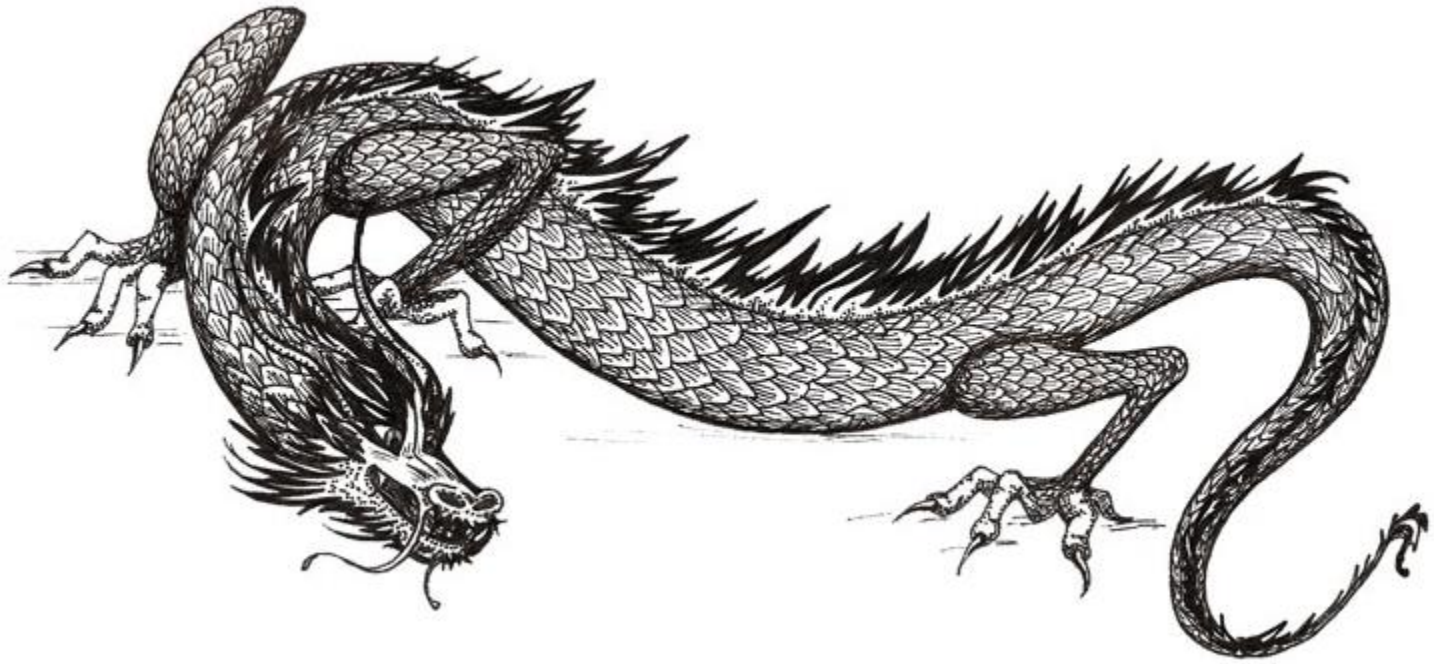


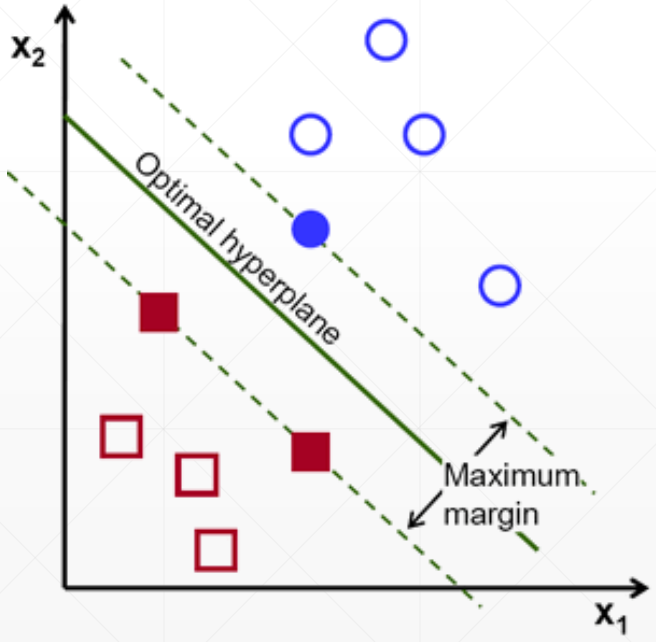
损失函数

主讲：龙良曲

Outline

- MSE
- Cross Entropy Loss
- Hinge Loss

$$\sum_i max(0, 1 - y_i * h_\theta(x_i))$$



MSE

- $loss = \frac{1}{N}\sum(y - out)^2$

- $L_{2-norm} = \sqrt{\sum(y - out)^2}$

```
y = tf.constant([1, 2, 3, 0, 2])
y = tf.one_hot(y, depth=4)
y = tf.cast(y, dtype=tf.float32)

out = tf.random.normal([5, 4])


loss1 = tf.reduce_mean(tf.square(y-out))

loss2 = tf.square(tf.norm(y-out))/(5*4)

loss3 = tf.reduce_mean(tf.losses.MSE(y, out)) # VS MeanSquaredError is a class

tf.Tensor(1.0918376, shape=(), dtype=float32)
tf.Tensor(1.0918376, shape=(), dtype=float32)
tf.Tensor(1.0918376, shape=(), dtype=float32)
```

Entropy

- Uncertainty
- measure of surprise

- lower entropy → more info.

$$Entropy = -\sum_i P(i) \log P(i)$$

Claude Shannon

https://towardsdatascience.com/demystifying-cross-entropy-e80e3ad54a8

Lottery

```
In [3]: a=tf.fill([4],0.25)
In [6]: a*tf.math.log(a)/tf.math.log(2.)
Out[6]: <tf.Tensor: id=13, shape=(4,), dtype=float32, numpy=array([-0.5, -0.5,
-0.5, -0.5], dtype=float32)>
In [7]: -tf.reduce_sum(a*tf.math.log(a)/tf.math.log(2.))
Out[7]: <tf.Tensor: id=22, shape=(), dtype=float32, numpy=2.0>

In [8]: a=tf.constant([0.1,0.1,0.1,0.7])
In [9]: -tf.reduce_sum(a*tf.math.log(a)/tf.math.log(2.))
Out[9]: <tf.Tensor: id=32, shape=(), dtype=float32, numpy=1.3567796>

In [10]: a=tf.constant([0.01,0.01,0.01,0.97])
In [11]: -tf.reduce_sum(a*tf.math.log(a)/tf.math.log(2.))
Out[11]: <tf.Tensor: id=42, shape=(), dtype=float32, numpy=0.24194068>
```

Cross Entropy

$$H(p,q) = -\sum p(x) \log q(x)$$

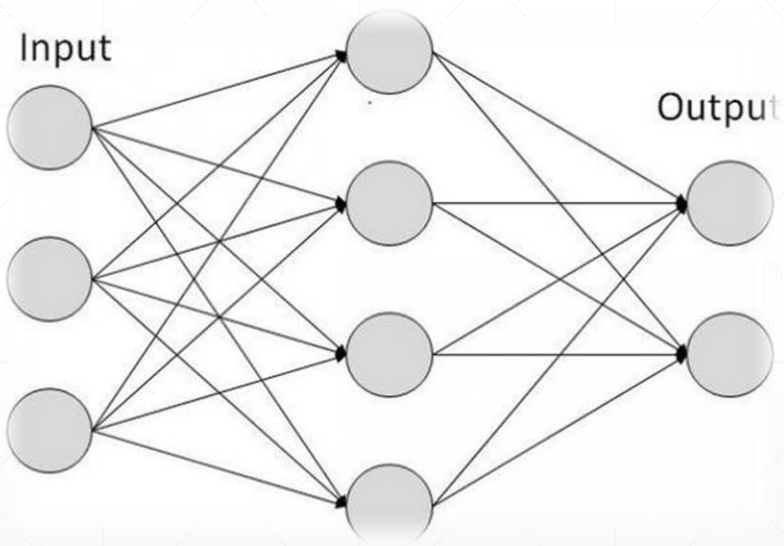
$$H(p,q) = H(p) + D_{\mathrm{KL}}(p|q).$$

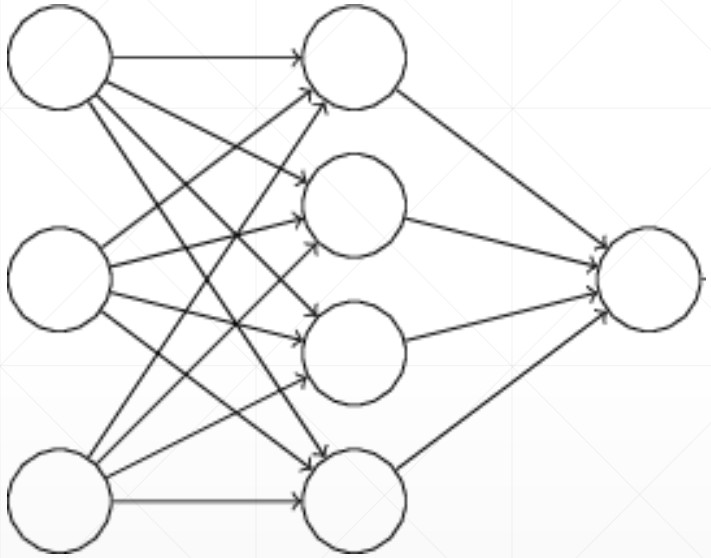
- for $p = q$
  - Minima: $H(p,q) = H(p)$

- for p: one-hot encoding
  - $h(p\colon [0,1,0]) = -1log1 = 0$
  - $H([0,1,0],[p_0,p_1,p_2]) = 0 + D_{KL}(p|q) = \boldsymbol{-1logq_1}$

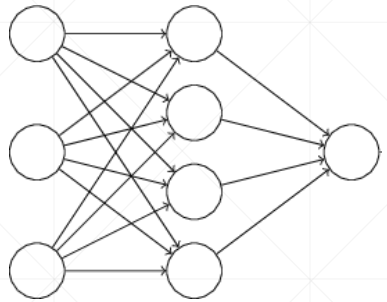
Binary Classification

- Two cases





Single output



$$H(P,Q) = -P(cat)\log Q(cat) - (1 - P(cat))\log(1 - Q(cat))$$

$$P(dog) = (1 - P(cat))$$

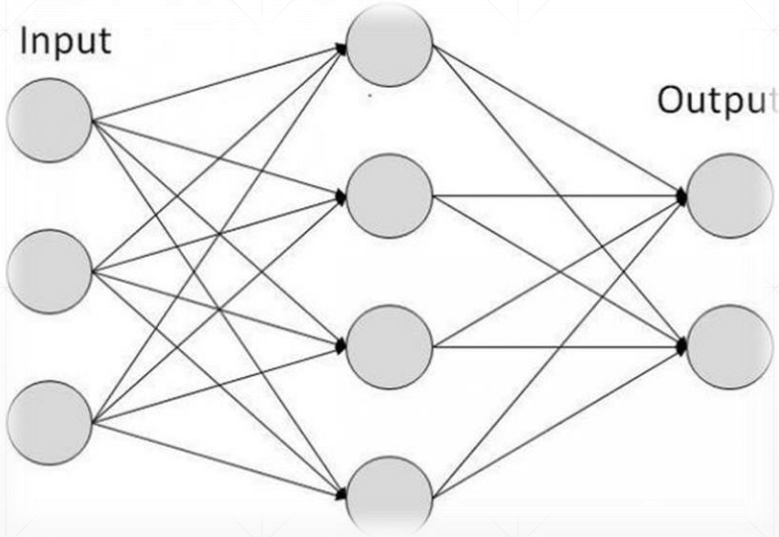
$$H(P,Q) = -\sum_{i=(cat,dog)} P(i)\log Q(i)$$

$$= -P(cat)\log Q(cat) - P(dog)\log Q(dog)$$

$$-(y\log(p) + (1-y)\log(1-p))$$

Classification

- $H([0,1,0],[p_0,p_1,p_2]) = 0 + D_{KL}(p|q) = -\mathbf{1}\boldsymbol{log}\boldsymbol{q_1}$



Classification

$$P_1 = [1 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$Q_1 = [0.4 \quad 0.3 \quad 0.05 \quad 0.05 \quad 0.2]$$

$$\begin{aligned} H(P_1, Q_1) &= -\sum_i P_1(i) \log Q_1(i) \\ &= -(1 \log 0.4 + 0 \log 0.3 + 0 \log 0.05 + 0 \log 0.05 + 0 \log 0.2) \\ &= -\log 0.4 \\ &\approx 0.916 \end{aligned}$$

$$Q_1 = [0.98 \quad 0.01 \quad 0 \quad 0 \quad 0.01]$$

$$\begin{aligned} H(P_1, Q_1) &= -\sum_i P_1(i) \log Q_1(i) \\ &= -(1 \log 0.98 + 0 \log 0.01 + 0 \log 0 + 0 \log 0 + 0 \log 0.01) \\ &= -\log 0.98 \\ &\approx 0.02 \end{aligned}$$

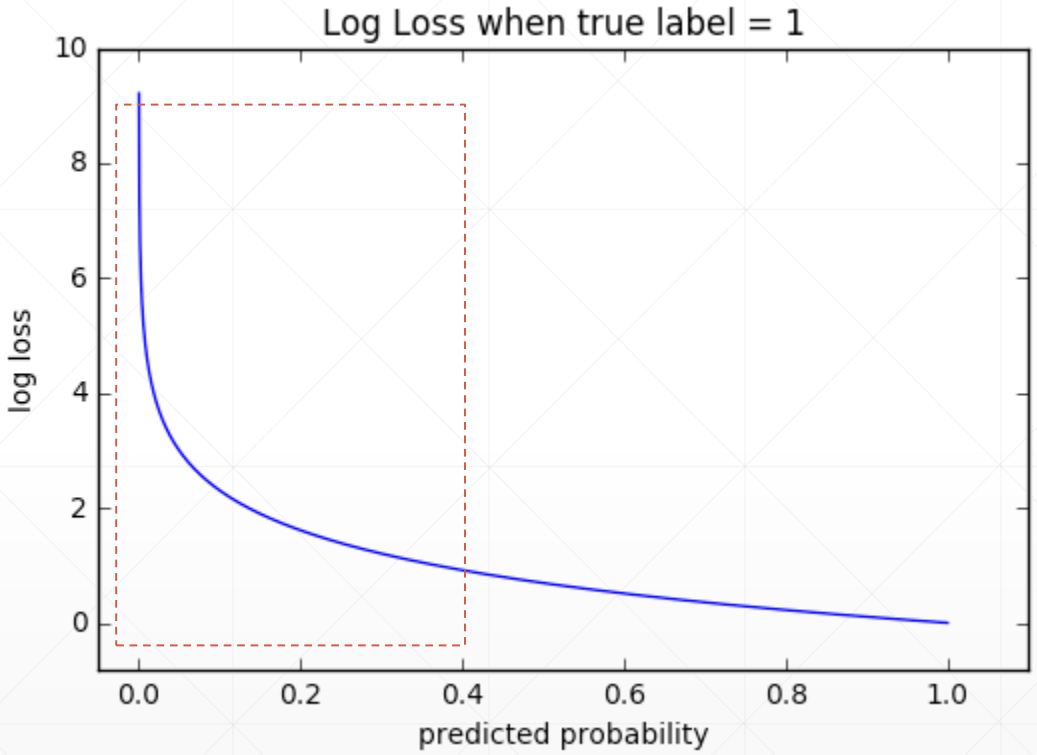
Categorical Cross Entropy

```
In [15]: tf.losses.categorical_crossentropy([0,1,0,0],[0.25,0.25,0.25,0.25])
Out[15]: <tf.Tensor: id=98, shape=(), dtype=float32, numpy=1.3862944>

In [16]: tf.losses.categorical_crossentropy([0,1,0,0],[0.1,0.1,0.8,0.1])
Out[16]: <tf.Tensor: id=117, shape=(), dtype=float32, numpy=2.3978953>

In [17]: tf.losses.categorical_crossentropy([0,1,0,0],[0.1,0.7,0.1,0.1])
Out[17]: <tf.Tensor: id=136, shape=(), dtype=float32, numpy=0.35667497>

In [18]: tf.losses.categorical_crossentropy([0,1,0,0],[0.01,0.97,0.01,0.01])
Out[18]: <tf.Tensor: id=155, shape=(), dtype=float32, numpy=0.030459179>
```

```
In [20]: criteon([0,1,0,0],[0.1,0.7,0.1,0.1])
Out[20]: <tf.Tensor: id=186, shape=(), dtype=float32, numpy=0.35667497>

In [21]: criteon([0,1],[0.9,0.1])
Out[21]: <tf.Tensor: id=216, shape=(), dtype=float32, numpy=2.3025851>

In [22]: tf.losses.BinaryCrossentropy()([1],[0.1])
Out[22]: <tf.Tensor: id=254, shape=(), dtype=float32, numpy=2.3025842>

In [23]: tf.losses.binary_crossentropy([1],[0.1])
Out[23]: <tf.Tensor: id=281, shape=(), dtype=float32, numpy=2.3025842>
```

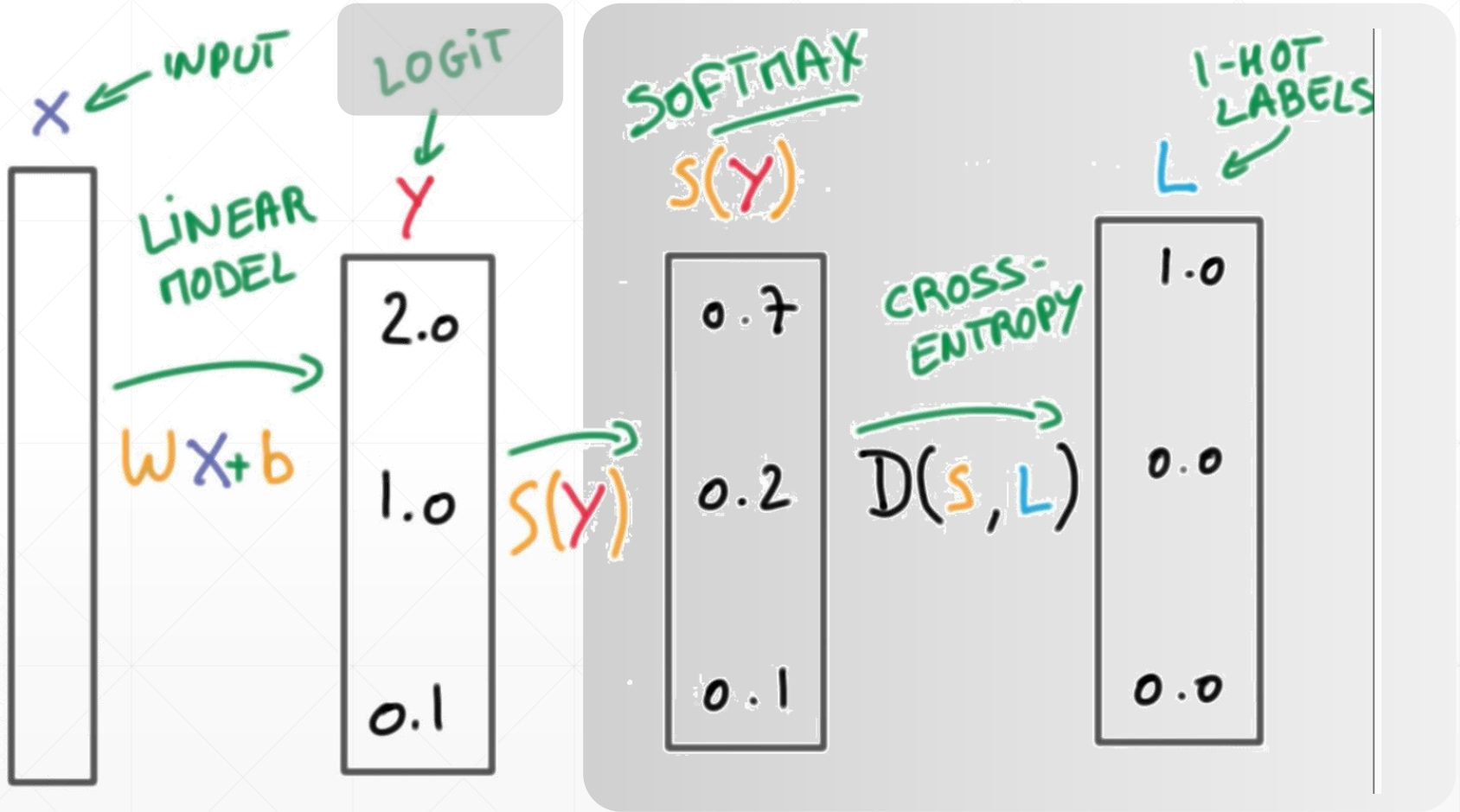
Why not MSE?

- sigmoid + MSE
  - gradient vanish
- converge slower
- However
  - e.g. meta-learning



https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html

logits→CrossEntropy



Numerical Stability

```
In [24]: x=tf.random.normal([1,784])
In [25]: w=tf.random.normal([784,2])
In [26]: b=tf.zeros([2])

In [27]: logits=x@w+b
Out[29]: <tf.Tensor: id=299, shape=(1, 2), dtype=float32, numpy=array([[-26.27812,
 28.63038]], dtype=float32)>

In [30]: prob=tf.math.softmax(logits,axis=1)
Out[31]: <tf.Tensor: id=301, shape=(1, 2), dtype=float32,
numpy=array([[1.4241021e-24, 1.0000000e+00]], dtype=float32)>

In [34]: tf.losses.categorical_crossentropy([0,1],logits, from_logits=True)
Out[34]: <tf.Tensor: id=393, shape=(1,), dtype=float32, numpy=array([0.],
dtype=float32)>

In [35]: tf.losses.categorical_crossentropy([0,1],prob)
Out[35]: <tf.Tensor: id=411, shape=(1,), dtype=float32, numpy=array([1.192093e-
07], dtype=float32)>
```

下一课时

梯度下降

Thank You.