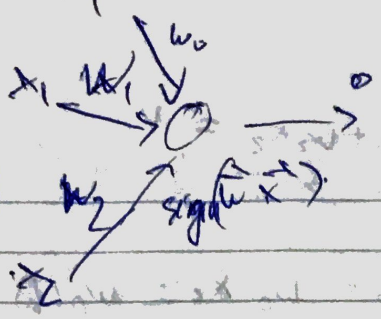
THOMAS NGO

8/21

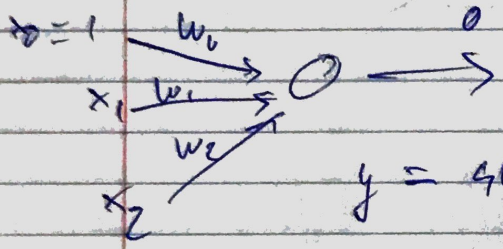


(1.(a)

$$\hat{y} = y = \text{sign}\{\vec{w}\cdot\vec{x}\} = \text{sign}\{\sum_{i=1}^{2} w_i x_i\} = 0 \text{ (or 1)}$$

Not Zero.

(b) $b = x_0 \cdot w_0 = 1 \cdot (0.1) = 0.1$ (Bias)



$$y = \text{sign}\{w_0 + w_1 x_1 + w_2 x_2\} = 0$$

$$= \text{sign}\{0.1 + 0.2(2) + (-0.3)\cdot 3\}.$$

(c) $\hat{y} = \text{sign}\{0.1 + 0.2(2) + (-0.3)\cdot 3\}.$

$= \text{sign}\{-0.5\} = -1.$

(d) $\mathcal{L} = (y - \hat{y})^2 = (1 - (-1))^2 = 2^2 = 4.$

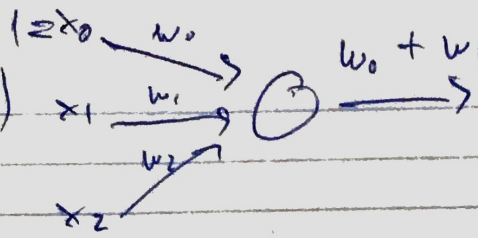
(e) $\vec{w} = \vec{w} + \alpha(y - \hat{y})\vec{x}$

$w = (0.1, 0.2, -0.3) + \alpha(4)(1, 2, 3).$

$= (0.1, 0.2, -0.3) + (4\alpha, 8\alpha, 12\alpha)$

$= (0.1 + 4\alpha, 0.2 + 8\alpha, -0.3 + 12\alpha)$

(2) (a)



$b = w_0 x_0 = w_0(1) = w_0$ (bias)

(b) $\Phi(v) = v$ (identity)

$\Phi(w_0 + w_1 x_1 + w_2 x_2) = w_0 + w_1 x_1 + w_2 x_2$.

(b) $\frac{\partial L}{\partial x_1} = \frac{\partial L}{\partial o} \cdot \frac{\partial o}{\partial x_1} = (5)(0.2) = 1$.

$\frac{\partial o}{\partial x_1} = w_1$.

(c) $\frac{\partial L}{\partial x_2} = \frac{\partial L}{\partial o} \cdot \frac{\partial o}{\partial x_2} = (5)(-0.3) = -1.5$

$\frac{\partial o}{\partial x_2} = w_2$

$\frac{\partial L}{\partial x_0} = \frac{\partial L}{\partial o} \cdot \frac{\partial o}{\partial x_0} = (5)(0) = 0$

$\frac{\partial o}{\partial x_0} = 0$.

$\Rightarrow \nabla_{\vec{w}} L = \begin{pmatrix} \frac{\partial L}{\partial w_0} \\ \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_2} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ -1.5 \end{pmatrix}$

(2)

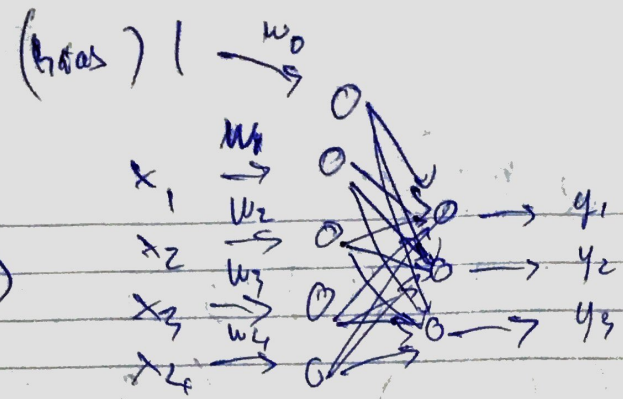
(d) $\vec{w} \leftarrow \vec{w} - \alpha \nabla_{\vec{w}} \mathcal{L}$

$w = (0.1, 0.2, -0.3) - (0.1)(0, 1, -1.5)$

$= \begin{pmatrix} 0.1 - 0.1(0) \\ 0.2 - 0.1(1) \\ -0.3 - 0.1(-1.5) \end{pmatrix} = \begin{pmatrix} 0 \\ 0.1 \\ -2.45 \end{pmatrix}$

(3)

(a)



4 inputs because there are 4 features

(b) 3 outputs because there are 3 different classes

(c) 15 weights (including bias)

(d) Multinomial logistic Regression is a Softmax classifier.
=> use Softmax activation and
use cross-entropy as loss function.

(e) Given $\vec{x}$, The Multinomial logistic regression will give us the outputs that contain likelihood or probabilities of each class - Based on the probability, the prediction is the highest likelihood/probability or most likely class to occur.

(4) (a) if α is too small, the learning rate becomes slow. => Takes longer time to train if we want to get high accuracy

(b) #'s of neuron in hidden layer is low -> The accuracy is low because model underfits training set

(c) # of epochs is low -> The learning may not completed or converged -> the accuracy is low and high loss

(d) if λ is small, then model cannot penalize the weights. Some of weights should not have high impact/influence on some features. The regularizer helps us to lower these influence.

(e) Mini-batch size is small -> The training cannot learn fast. The group of inputs can be trained at the same is small -> learn slow and may effect accuracy.

(5) Some of potential causes:

- Use too much dense layers.

(a) This can have negative effect like overfitting the training set.

(b) They can use fewer #s of dense layers.

(c) if they change as suggested in (b), the accuracy and loss may get improved.

- Use inappropriate activation function and loss functions.

(a) This will cause the wrong prediction or unexpected result.

(b) they can change to different activation functions and loss functions

(c) if they apply the change, the outputs can be better.

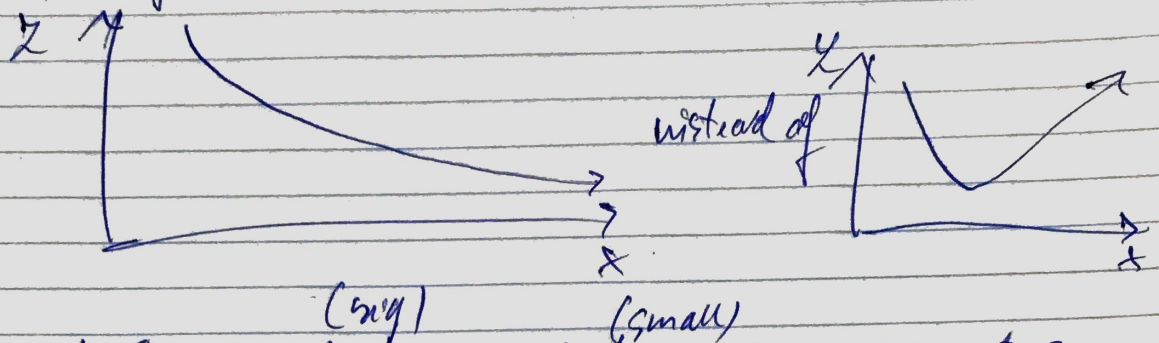
- Training data set may be too small.

(a) This can not make the model get information, learn the pattern and leads to approximate or overfitting.

(b) They can change to larger training set.

(c) if they apply the change, they would expect a better result. (e.g. higher accuracy, lower loss)

5 (c) They would expect something like this on graph.



← use ~~hyperp~~ very high or very low hyperparameters.

(a) This can cause the learning rate (α) to "learn" too fast or too slow. OR even worse, the training will never converge or learn nothing.

(b) They can tune hyperparams to the appropriate values

(c) They would expect. the see the loss as ~~fo~~ picture above on the left.