The wordle game is a popular online word game where players have six attempts to guess a five-letter word. After each guess, the game gives feedback in the form of coloured tiles.

- Green for a correct letter in the correct position.
- Yellow for a correct letter in the wrong position
- Gray for a letter not in the word at all.

To apply the knowledge of information theory to solve the wordle game, one would focus on maximizing information gain with guess. A key concept entropy, which measures the uncertainty or information content. In the wordle, each guess should reduce the entropy by providing information that narrow down the set of possible words.
Lets say, we have 5757 words in English language.
So, for each word we need $\log_2(5757)$ or 12.49 or 13 bits of information but the rule of the game is to guess in 6 bits of information.

Let our initial guess be 'tares'
There are 12 words in English language that has the 4 correct letters. Now the uncertainty has reduced from 12.49 to $\log_2(12)$ or 3.58. So, first guess made us gain 8.91 bits of information. Now, if we guess the word 'orate'. the only possible word that remains is our answer 'great'

great
⋮
~~[illegible]~~
heart
areto
⋮
orate
⋮

} 12 words

Now, what if we don't know the result. Then should we have guessed the initial guess.

So, let's say, initial guess was randomly taken and it was fuzzy. It was seen that 3543/5757 or 62% of the five letter words doesn't contain the letters of fuzzy. So, we need to guess a word that will give us highest information gain. For this reason, we can guess 'tares', as initial guess, it will have only 7% chance of giving us all grey. So, Information gain is much more than the fuzzy, which is 6.21 bits.

Information Vs Entropy

1.Information:

- In the context of information theory, developed by Claude Shannon, information is quantified as a measure of surprise or uncertainty. The more surprising or uncertain an event, the more information it provides.

- Information is often measured in bits. One bit represents the binary choice between two equally likely outcomes (such as true/false or 0/1).

2. Entropy:

- Entropy, also introduced by Claude Shannon in information theory, is a measure of the average amount of information or uncertainty associated with a set of possible outcomes.

- It is a concept borrowed from thermodynamics and has been adapted to quantify information. In information theory, entropy is used to measure the randomness or disorder in a system.

- Higher entropy indicates greater disorder or unpredictability, while lower entropy indicates more order or predictability.

- Entropy is used in the context of probability distributions. For example, a fair coin has higher entropy than a biased coin because the outcomes are more uncertain with the fair coin.

The relationship between information and entropy can be summarized as follows:

- High entropy: When a system has high entropy, there is greater unpredictability or disorder. This implies that each new piece of information about the system provides more "surprise" or "newness," and therefore, it carries more information.

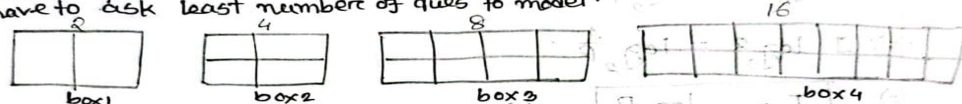
- Low entropy: Conversely, when a system has low entropy, there is more predictability or order. In this case, additional information about the system may not be as surprising or informative because the outcomes are more certain.

In summary, while information measures the content or surprise of a message, entropy quantifies the uncertainty or disorder in a set of possible outcomes. They are interconnected concepts used in information theory to analyze and quantify the characteristics of data and communication systems.

Q1. Write down your ideas about information theory. How can the logistics loss function be derived using information theory?

Determine how information and probability relate to one another. How can the logistics loss function be derived using information theory?

Suppose there some boxes with balls & have to guess/predict which portion of the boxes have the ball. To predict ball have to ask least number of ques to model.



q: 1 q: 2 q: 3 q: 4

Probability:

$P=\frac{1}{2}$ $P=\frac{1}{4}$ $P=\frac{1}{8}$ $P=\frac{1}{16}$

$=\frac{1}{2^1}$ $=\frac{1}{2^2}$ $=\frac{1}{2^3}$ $=\frac{1}{2^4}$ $\cdots\cdots \frac{1}{2^n}$

Here to find ball at least 1 ques need to get/predict the ans for box 1. 2, 3, 4 for box2, box3, box4 respectively. Strategy is half the box in every possible portion.

Now, here ques means gathering Information so for box1 need 1 information to predict and so on. & there probality is $P=\frac{1}{2}=\frac{1}{2^1}=\frac{1}{2^I}=\left(\frac{1}{2}\right)^I$

From this,

$\cancel{\left(\frac{1}{2}\right)^I = P}$

$\cancel{\Rightarrow \log_2\left(\frac{1}{2}\right)^I = \log_2 P}$

$\cancel{\Rightarrow I \log}$

From this,

$$\left(\frac{1}{2}\right)^I = p$$

$$\Rightarrow 2^I = \frac{1}{p}$$

$$\Rightarrow \log_2 2^I = \log_2 \frac{1}{P}$$

$$\Rightarrow I \log_2 2 = \log_2 \frac{1}{P}$$

$$\Rightarrow \boxed{I = -\log_2 P} \quad \text{—(1)}$$

Hence, thus creates ~~the~~ relation betn Information & Probability.

From equation (1) multiplying True value of data we get,

$$I = -t \log_2 P$$

Now, Let An animal is predic

	cat	dog	
Prob:	0.7	0.3	= (1 - 0.7)
	1	0	

So, Information theory can be written,

$$I = -t_c \log_2 P_c - (1 - t_c) \log_2 (1 - P_c)$$

$$\Rightarrow J = -y \log_2 \hat{y} - (1-y) \log_2 (1-\hat{y})$$

which is ~~equa~~ Loss function of Logistic regress / binary classi

2.Write down the basic idea of Entropy. What is the difference between Information and Entropy?

Ans: Entropy of a random variable X is the level of uncertainty inherent in the variables possible outcome.

1.Information:

- In the context of information theory, developed by Claude Shannon, **information is quantified as a measure of surprise or uncertainty. The more surprising or uncertain an event, the more information it provides.**

- Information is often measured in bits. One bit represents the binary choice between two equally likely outcomes (such as true/false or 0/1).

2. Entropy:

- Entropy, also introduced by Claude Shannon in information theory, **is a measure of the average amount of information or uncertainty associated with a set of possible outcomes**.

- It is a concept borrowed from thermodynamics and has been adapted to quantify information. In information theory, **entropy is used to measure the randomness or disorder in a system.**

- Higher entropy indicates greater disorder or unpredictability, while lower entropy indicates more order or predictability.

- Entropy is used in the context of probability distributions. For example, a fair coin has higher entropy than a biased coin because the outcomes are more uncertain with the fair coi

3.Derive the binary cross entropy loss function from the general formula with proper notation.

Q: Derive the binary Cross Entropy loss function from the general formula with proper notation.

Ans: The cross entropy loss function is called logarithmic loss, log loss or logistic loss.

The general formula for n classes,

$$L = -\sum_{i=1}^{n} t_i \log(P_i)$$

Here, t_i is the true label and P_i is the softmax Probability for i^{th} class.

Now, Let's consider binary classification scenario where cat (C) & dog (D) two classes. So in general it can be written,

$$L = -\left[t_c \log(P_c) + t_D \log(P_D)\right]$$

Since there is two classes, so one class can be written base on other,

cat 0.7 dog 0.3 = (1-0.7)

$$L = -\left[t_c \log(P_c) + (1-t_c) \log(1-P_c)\right]$$

$$L(y,\hat{y}) = -\left[y \log_2(\hat{y}) + (1-y) \log_2(1-\hat{y})\right]$$

where $\hat{y}$ represents the predicted probability belongs to class 1

& $(1-\hat{y})$ represents the predicted probability belongs to class 0

$$L = -\sum_{i=1}^{2} y_i \log(P_i)$$

4.Write down the difference between loss and cost function with proper equations.

Ans: Let's define each and highlight the differences:

1.Loss Function:

- The loss function is typically associated with a single data point in a dataset. It measures the error between the predicted output and the true target for that specific data point.

- Denoted by (L(hat{y}, y)), where (hat{y}) is the **predicted output** and (y) is the true target.

- Common examples include Mean Squared Error (MSE), Binary Cross-Entropy Loss, and Categorical Cross-Entropy Loss.

- For a single data point, the loss function might look

$L(y\^{}, y) = -[y \log(y\^{}) + (1-y)\log(1-y\^{})]$ for Binary Cross-Entropy Loss.

2. Cost Function:

- The cost function, on the other hand, is the average loss over the entire dataset. It is the overall measure of how well the model is performing on the entire training set.

- It is the sum (or average) of the individual loss functions over all training examples.

- For a dataset of size \(N \), the cost function might be defined as

$$L = -\frac{1}{N}\left[\sum_{j=1}^{N}[t_j \log(p_j) + (1 - t_j)\log(1 - p_j)]\right]$$

where pj is the predicted output for the jth data point and Tj is the true target.

In summary, the loss function quantifies the error for a single data point, while the cost function represents the overall performance of the model on the entire dataset. The cost function is essentially an average or sum of the losses across all data points.

5.Why do we use softmax as opposed to standard normalization? Give proper justification of your answer.

Why do use softmax as opposed to standard normaliza

Ans: The softmax function is particularly well-suited for ~~this~~ classification class than standard normalization because it not ~~no~~ only normalize the output to be in the range (0,1) but also ensure that the outputs sum to 1 and also ~~capble~~ capable to capture the changes in input features to the way of output.

For example; Let's take a blurry image of a ferret

Now softmax [1,2] = $\frac{e^1}{e^1+e^2}$, $\frac{e^2}{e^1+e^2}$

= 0.2689 , 0.7310

It is hard to identify as cat. But if we take better crisp resulation image of cat about [10,20]

Softmax [10,20] = $\frac{e^{10}}{e^{10}+e^{20}}$, $\frac{e^{20}}{e^{10}+e^{20}}$

= 0.000045 , 0.99995

it is so easy to identify as Cat.

But for normalization, varying input feature doesnot change the output.

Std-norm [1,2] = $\frac{1}{1+2}$, $\frac{2}{1+2}$

= 0.333 , 0.666

Std-norm [10,20] = $\frac{10}{10+20}$, $\frac{20}{10+20}$

= 0.333 , 0.666

That's softmax opposed to standard normalization.

QUIZ-2

Q1. Answer the following questions.

1.Describe the role of a ReLU layer in a convolutional neural network. [3]

Ans: The role of a ReLU layer in a CNN includes the following:

a. Non-Linearity: The ReLU layer adds a touch of complexity to the network by making sure that, if the input is positive, it stays as it is, but if it's negative, it turns into zero. This introduces non-linearity, allowing the network to learn and understand more complicated patterns in data.

b. Feature Map Sparsity: ReLU helps create sparse feature maps by zeroing out negative values. This sparsity can be beneficial as it encourages the network to focus on more important features, improving generalization and reducing the risk of overfitting.

c. Gradient Descent Optimization: ReLU facilitates gradient-based optimization during backpropagation. Its derivative is either 0 or 1, making it computationally efficient for calculating gradients

"it is important to note that ReLU can suffer from the "dying ReLU" problem, where neurons may become inactive during training and stop learning."

d. Improved Training Convergence: The non-saturation property of ReLU (unlike some other activation functions that saturate for large positive or negative inputs) can lead to faster convergence during training, as it mitigates the vanishing gradient problem in deep networks.

2."A neural network is a combination of multiple logistic regressions." - Justify the statement. [7]

Q: "A neural network is a combination of multiple logistic regressions." justify the statement.

Ans: This statement ~~is~~ can be justified by describing the basic structure ~~of~~ and functioning of a neural network and logistic regression.

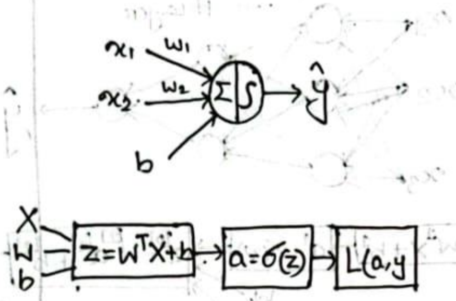


Figure:1 Logistic Regression

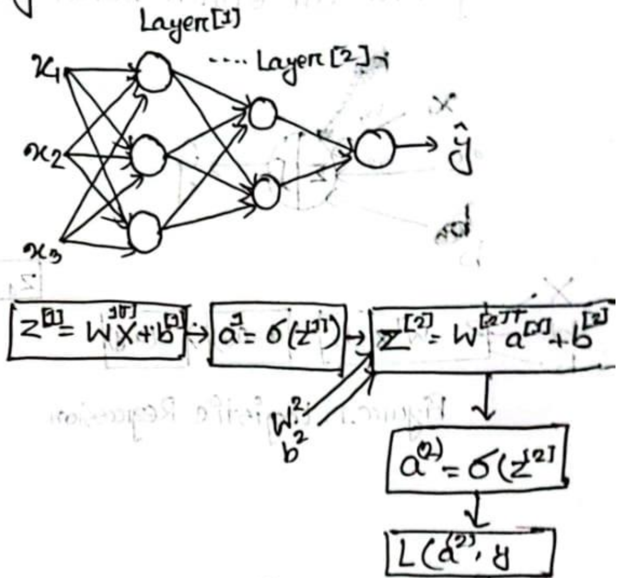


Figure2: Neural Network.

In logistic Regression, the input features are linearly combined and output is obtained by applying the sigmoid activation function to the weighted sum of inputs. And then calculate loss, which is visually represented in Figure1. LR is consider as single neuron.

And a neural network is nothing but a stacking up multiple logistic regression and consisting of layers of interconnected units (LR) organized in an input layers, ~~and~~ one or more ~~or~~ hidden layers and output layer. The outputs of these units from one layer serve as inputs to the next layer, followed by an activation function, figure 2.

$$\text{Output}_{(\text{neural Network})} = \text{Activation}\left(\sum_i \left(\text{weight}_{ij} \times \text{output}_{LR_i}\right) + \text{Bias}_j\right)$$

So, the key point is that each neuron in a layer operates similarly to a logistic regression unit and the combination of these unit across layers form the structure of a neural network. So LR can b considered as the simplest form of neural networ

And thus the given statement is justified.

In essence, a neural network can be seen as a composition of multiple logistic regressions, but the power and expressiveness of a neural network arise from the combination of these basic units and the introduction of non-linear activation functions, allowing it to model complex relationships in data.

Q1. Answer the following questions.

1.Describe the role of a convolution layer in a convolutional neural network. [3]

In a Convolutional Neural Network (CNN), a convolutional layer plays a crucial role in capturing patterns and features from input data, such as images. Here's a simple description:

1. Feature Extraction:

- The convolutional layer scans the input image with small filters (also known as kernels or convolutions).

- These filters capture local patterns or features, like edges, corners, or textures, in the image.

2. Spatial Hierarchy:

- By using multiple filters, the convolutional layer creates a spatial hierarchy of features, **learning to recognize more complex patterns as it goes deeper into the network.**

3. Parameter Sharing:

- Parameters (weights) of the filters are shared across the entire image, reducing the number of parameters and making the network more efficient.

4. Translation Invariance:

- Convolutional layers introduce translation invariance, meaning the network can recognize patterns regardless of their exact position in the image.

In simpler terms, think of a convolutional layer as a feature detector that slides over an image, recognizing different shapes and patterns. This helps the neural network understand and learn hierarchical representations of visual features, making it effective for tasks like image recognition.

2. "The derivative of the tanh activation function depends on the function itself." - Justify the statement. [7]

Q: "The derivative of the tanh activation function depends on the function itself". Justify the statement.

Ans: We know, $\tanh = \dfrac{e^z - e^{-z}}{e^z + e^{-z}}$ $\qquad \boxed{\dfrac{u}{v} = \dfrac{vu' - uv'}{v^2}}$ will be.

If tanh is an activation function then derivation of

$$\frac{\partial \tanh}{\partial z} = \frac{\partial}{\partial z}\left[\frac{e^z - e^{-z}}{e^z + e^{-z}}\right]$$

$$= \frac{(e^z + e^{-z})(e^z + e^{-z}) - (e^z - e^{-z})(e^z - e^{-z})}{(e^z + e^{-z})^2}$$

$$= \frac{(e^z + e^{-z})^2 - (e^z - e^{-z})^2}{(e^z + e^{-z})^2}$$

$$= \frac{(e^z + e^{-z})^2}{(e^z + e^{-z})^2} - \left(\frac{e^z - e^{-z}}{e^z + e^{-z}}\right)^2$$

$$= 1 - (\tanh)^2$$

~~So~~ the derivation turns into $(1-\tanh^2)$ where except tanh no other variable is existed. Hence it is visiable that derivation of tanh only depends on itself.

Q1. Answer the following questions.

1."The derivative of the hyperbolic tangent function is more steep than the sigmoid function"- Justify the statement with proper evidence.

Q: "The derivative of the hyperbolic tangent function is more steep than the sigmoid function" - Justify the statement with proper evidance.

Ans: We know, for sigmoid activation function if large value is assigned then the gradient becomes zero same goes for very small value.

Now the sigmoid function, $g(z) = \frac{1}{1+e^{-z}}$

And the derivation of sigmoid is,

$g'(z) = g(z).(1 - g(z))$

Let's, if $z = 10$; $g(z) = \frac{1}{1+e^{-10}} = 0.999 \approx 1$

So $g'(z) = 1(1-1) = 0$ ✓

Again, if $z = -10$; $g(z) = \frac{1}{1+e^{-(-10)}} = 0.000045 \approx 0$

So, $g'(z) = 0(1-0) = 0$ ✓

For $z = 0$; $g(z) = \frac{1}{1+e^{-0}} = \frac{1}{2}$

So, $g'(z) = \frac{1}{2}(1-\frac{1}{2}) = \frac{1}{4} = 0.25$ ✓

So it is proved that for large and low value sigmoid shows vanishing gradient problem and

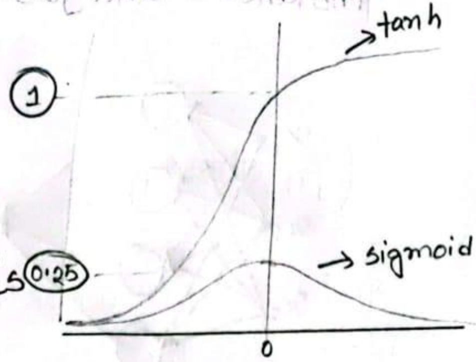
It's steepness is about 0.25 unit long if it is draw on graphically.

Now for tanh,

tanh, $g(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$

The derivation of tanh(z) is

$$g'(z) = 1 - (g(z))^2$$



Let's if $z = 10$; $g(z) = \frac{e^{10} - e^{-10}}{e^{10} + e^{-10}} \approx 0.9999 \approx 1$.

so $g'(z) = 1 - (1)^2 = 0$ ✓

Again if $z = -10$; $g(z) = \frac{e^{-10} - e^{-(-10)}}{e^{-10} + e^{-(-10)}} \approx -1$

so $g'(z) = 1 - (-1)^2 = 0$ ✓

Now, $z = 0$; $g(z) = \frac{e^{0} - e^{-0}}{e^{0} + e^{-0}} = 0$

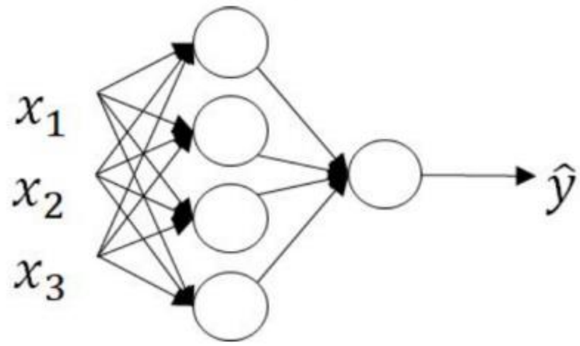
so $g'(z) = 1 - 0^2 = 1$ ✓

Tanh also suffers from vanishing G. problem but it's steepness much higher than sigmoid about 1.

Therefore the statement is justified.

Fig 1: Representation of steepness of tanh & sigmoid

Q1. Vectorize the following neural network for multiple instances and justify your implementation.



Q1: Vectorize the following neural network for multip instances and justify your implementation.

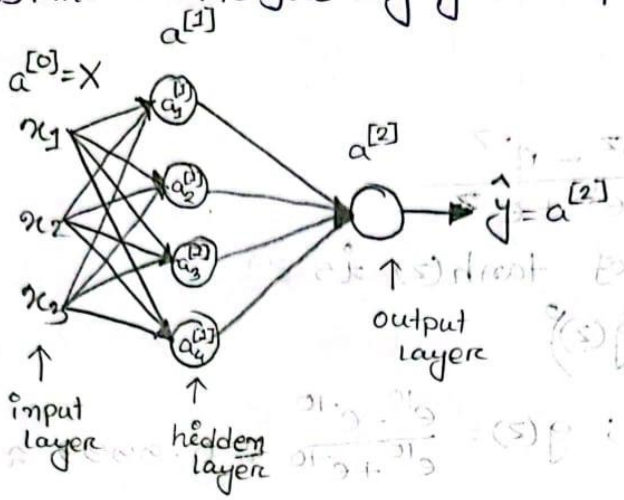


Figure 1 : Neural Network.

Neural network is nothing but stackup of multiple logistic regression figure:2

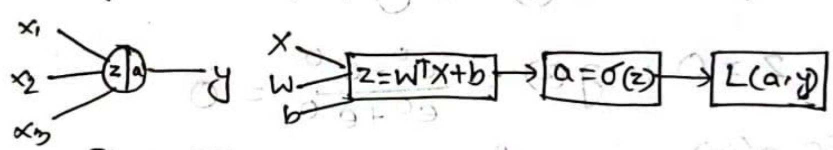


Fig2: LR

So neural network is all about vertically stack up each logistic regression/neuron and continue this repeated task for every hidden layer.

<u>For single instance. NN be like, Layer-1</u>

$$z_1^{[1]} = W_1^{[1]T} x + b_1^{[1]} \quad , \quad a_1^{[1]} = \sigma(z_1^{[1]})$$

$$z_2^{[1]} = W_2^{[1]T} x + b_2^{[1]} \quad , \quad a_2^{[1]} = \sigma(z_2^{[1]})$$

$$z_3^{[1]} = W_3^{[1]T} x + b_3^{[1]} \quad , \quad a_3^{[1]} = \sigma(z_3^{[1]})$$

$$z_4^{[1]} = W_4^{[1]} x + b_4^{[1]} \quad , \quad a_4^{[1]} = \sigma(z_4^{[1]})$$

Vectorize Representation:

$$Z^{[1]} = \begin{bmatrix} W_1^{[1]T} \cdots \\ W_2^{[1]T} \cdots \\ W_3^{[1]T} \cdots \\ W_4^{[1]T} \cdots \end{bmatrix} \cdot \begin{bmatrix} \vdots & & \vdots \\ x_1^{(1)} & \cdots & x_1^{(m)} \\ x_2^{(1)} & \cdots & x_2^{(m)} \\ \vdots & & \vdots \end{bmatrix} + \begin{bmatrix} b^{[1]} \\ \vdots \\ \vdots \end{bmatrix}$$

$$= \begin{bmatrix} W_1^{[1]} x^{(1)} & \cdots & W_m^{[1]}(x^{m}) \\ \vdots & & \\ W_4^{[1]} x^{(1)} & \cdots & W^{[1]} x^{(m)} \end{bmatrix} = \begin{bmatrix} z^{[1](1)} & z^{[1](2)} & \cdots & z^{[1][m]} \\ \vdots & \vdots & & \vdots \end{bmatrix} = z^{[1]}$$

Same for Activation

$$A^{[1]} = \begin{bmatrix} a^{[1](1)} & a^{[1](2)} & a^{[1](3)} & \cdots & a^{[1](m)} \\ \vdots & \vdots & \vdots & & \vdots \end{bmatrix}$$

So ~~row~~ each ~~col~~ ~~row~~ col consist in $Z^{[1]}$ hold Logits; and $A^{[1]}$'s each ~~col~~ ~~row~~ col hold the feature probability of data from 1 to m.

Thus applying vectorization on each layer will reduce the number of line of code but the computational cost remains unchanged.

Scanned with CamScanner

Vectorization of LR: for (multiple instance)

$\boxed{J=0;\ \boxed{dW1=0;\ dW2=0}\ ;\ db=0}$ → To store all dW together initialize vector $dW =$ np.zeros((nx,1)).

$\boxed{W1=0;\ W2=0}$; $b=0$ → Vector, $W =$ np.zeros.

Also create vector for Z, A.

Forward Pass:

Previously

1. For $i = 1$ to m
2. $Z^{(i)} = W1 * x1(i) + W2 * x2(i) + b$
3. $a^{(i)} = \text{sigmoid}(z(i))$
4. $J += -(y(i) * \log(a(i)) + (1-y(i)) * \log(1-a(i)))$

For line 2 →

$$W^T X = [W^T]_{1\times n} \begin{bmatrix} \vdots & \vdots & & \vdots \\ x^{(1)} & x^{(2)} & \cdots & x^{(m)} \\ \vdots & \vdots & & \vdots \end{bmatrix}_{(n\times m)} + [b\ b \cdots b]$$

$$= \left[W^T x^{(1)} + b \quad W^T x^{(2)} + b \quad \cdots \quad W^T x^{(m)} + b\right]_{(1\times m)}$$

$$= \left[z^{(1)} \quad z^{(2)} \quad \cdots\cdots \quad z^{(m)}\right]_{1\times m} = Z$$

$$= \left[a^{(1)} \quad a^{(2)} \quad \cdots\cdots \quad a^{(m)}\right]$$

$$= A = \sigma(Z)$$

So, $z^{(1)} = W^T x^{(1)} + b \qquad z^{(2)} = W^T x^{(2)} + b \qquad z^{(3)} = W^T x^{(3)} + b$

$a^{(1)} = \sigma(z^{(1)}) \qquad a^{(2)} = \sigma(z^{(2)}) \qquad a^{(3)} = \sigma(z^{(3)})$

Gradient Computation: (Back Propagation)

$dz^{(1)} = a^{(1)} - y^{(1)} \quad dz^{(2)} = a^{(2)} - y^{(2)}$

$dZ = [dz^{(1)} \; dz^{(2)} \cdots dz^{(m)}]$

Previously: $dw1 += dz(i) * x1(i)$

$dw2 += \cdots\cdots$

$dw = 0$

$~~db~~ = 0$

$dw += X^{(1)} dz^{(1)}$

$\vdots$

$dw += X^{(m)} dz^{(m)}$

$dw /= m$

modified

$dw = \frac{1}{m} X dZ^T$

$$dw = \frac{1}{m}\begin{bmatrix} \vdots & & \vdots \\ x^{(1)} & \cdots & x^{(m)} \\ \vdots & & \vdots \end{bmatrix}\begin{bmatrix} dz^{(1)} \\ dz^{(2)} \\ dz^{(3)} \\ dz^{(m)} \end{bmatrix}$$

$$dw = \frac{1}{m}\left[X^{(1)} dz^{(1)} + X^{(2)} dz^{(2)} + \cdots X^{m} dz^{(m)}\right]$$

$db = 0$

$db += dz^{(1)}$

$\vdots$

$db += dz^{(m)}$

$db /= m$

$db = \frac{1}{m} \sum_{i=1}^{m} dz(i)$

$db = \frac{1}{m}\, np.sum(dZ)$.