BACKPROPAGATION

The backpropagation algorithm is a fundamental component of neural networks, essential for training models effectively. It utilizes the chain rule to adjust the weights and biases of a network

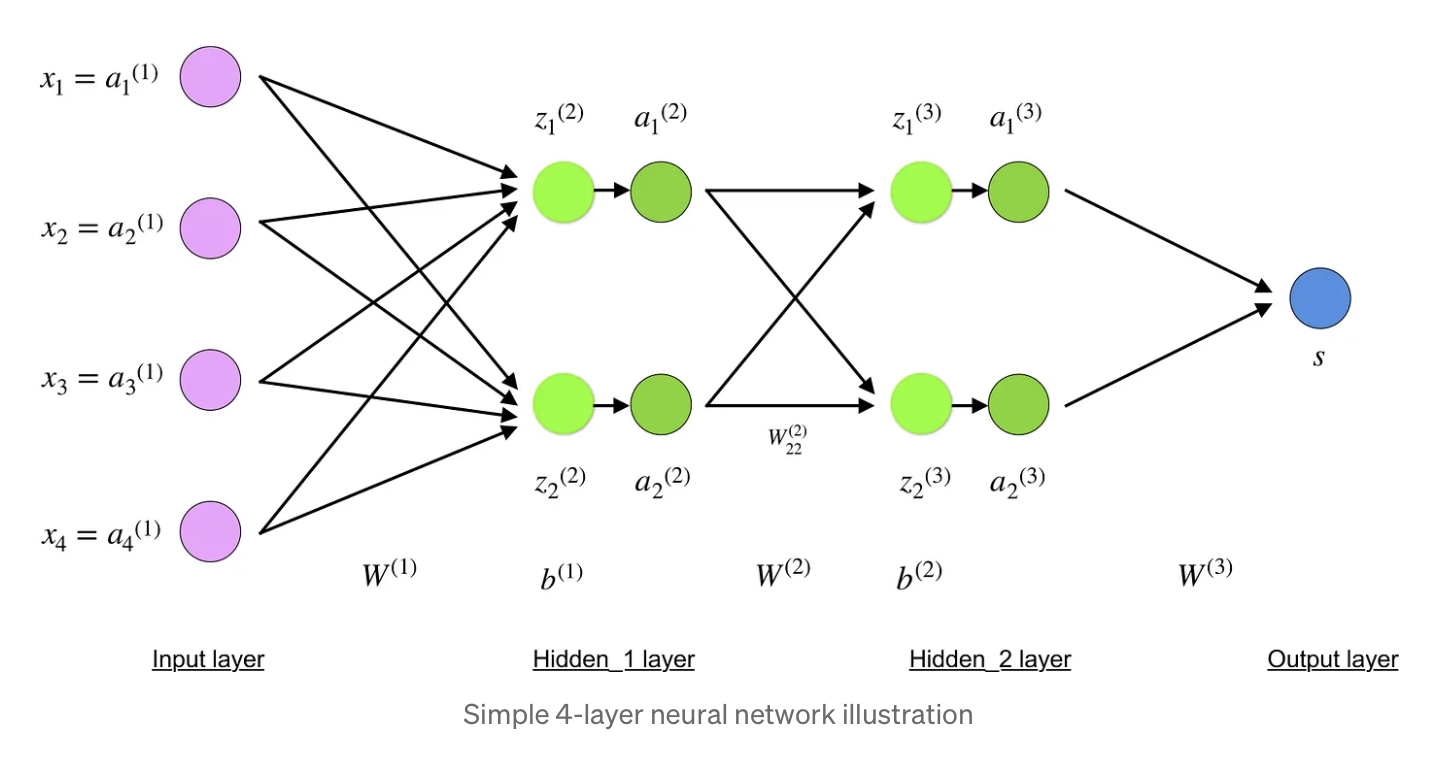
during backward passes following forward passes.

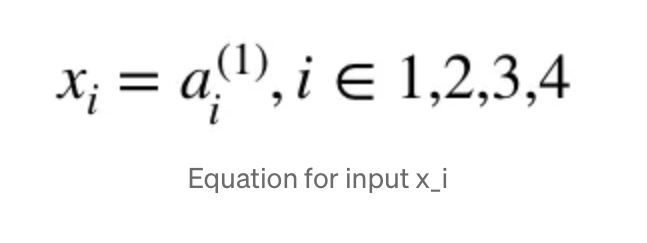
\*Neural network model

A 4-layer network is given -

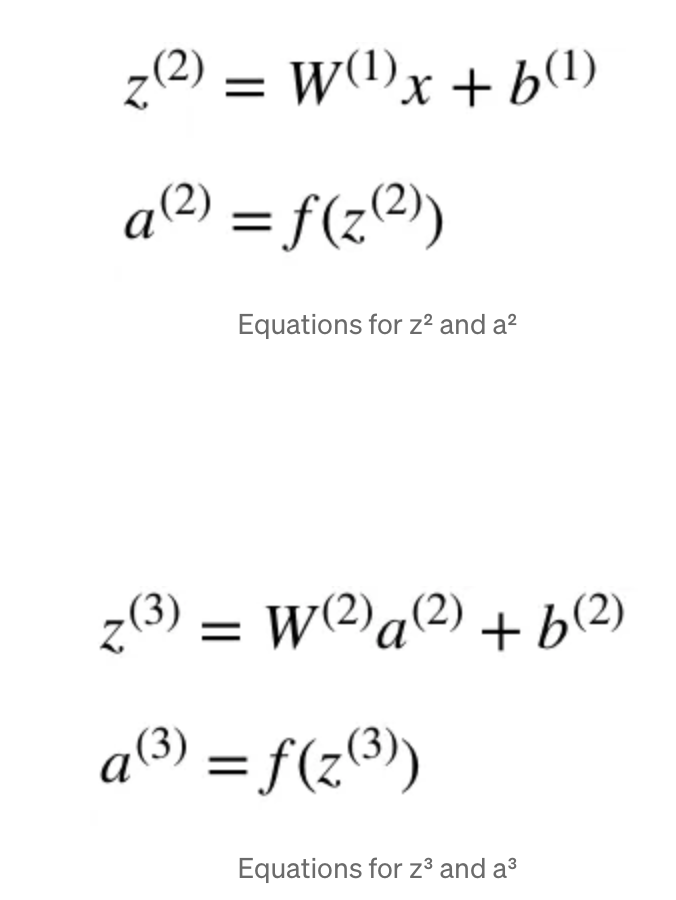
4 neurons - input layer

4 neurons - hidden layer



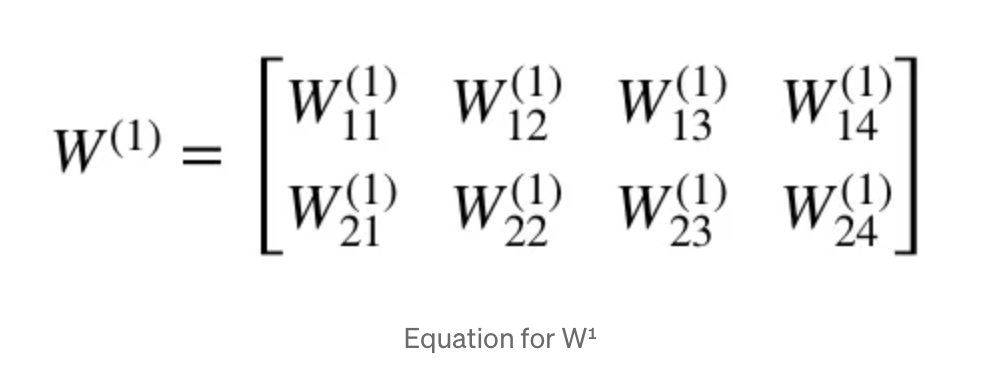
**Input layer-**

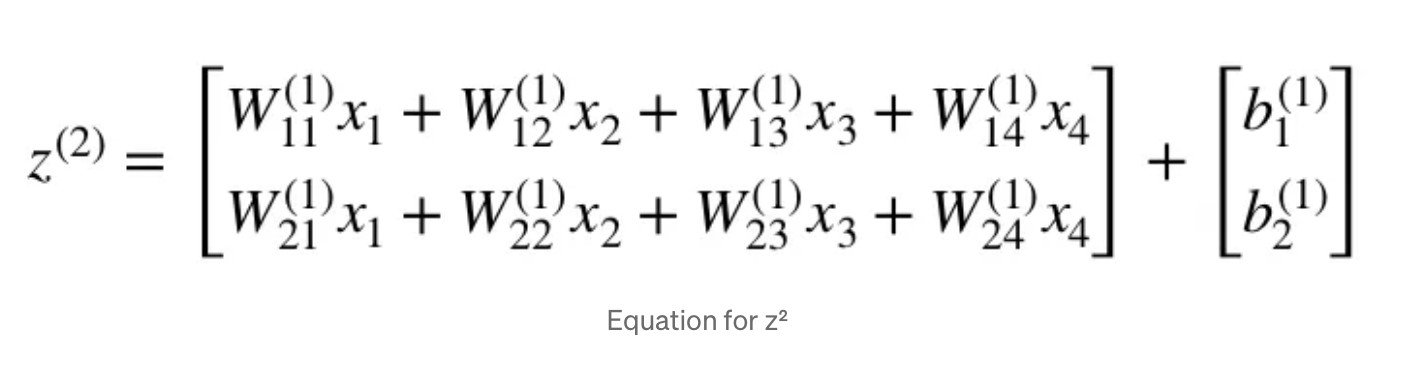
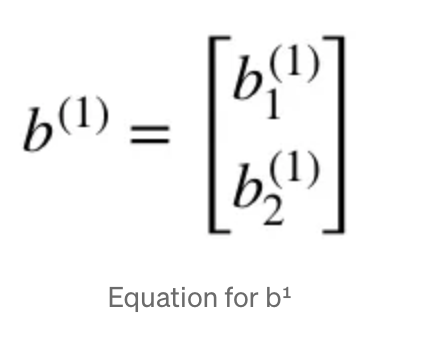
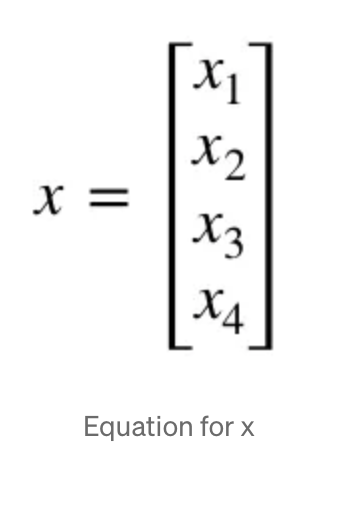
**Hidden layers-**

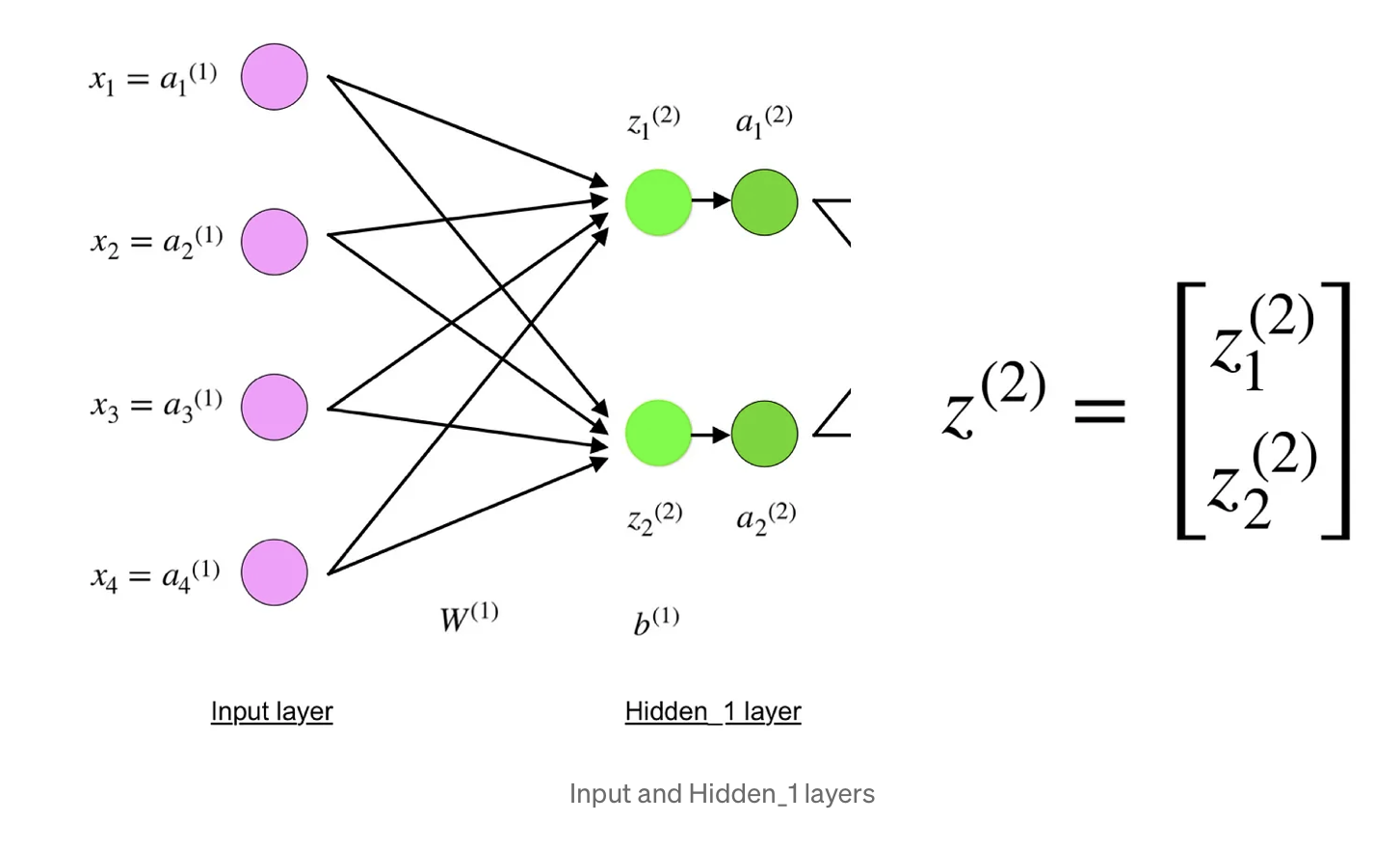
The final values at the hidden neurons, are computed using z^l — weighted inputs in layer l, and a^l— activations in layer l.

W² and W³ are the weights in layer 2 and 3 while b² and b³ are the biases in those layers.

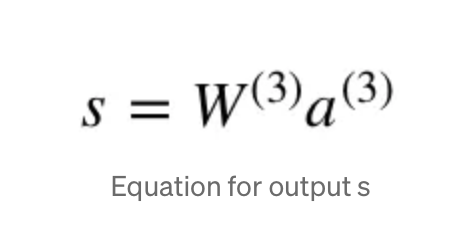
Lets take layer 2 as an example-

Layer 2 illustrates the weight matrix W¹, structured as (n, m) where n (output neurons) = 2 and m (input neurons) = 4.

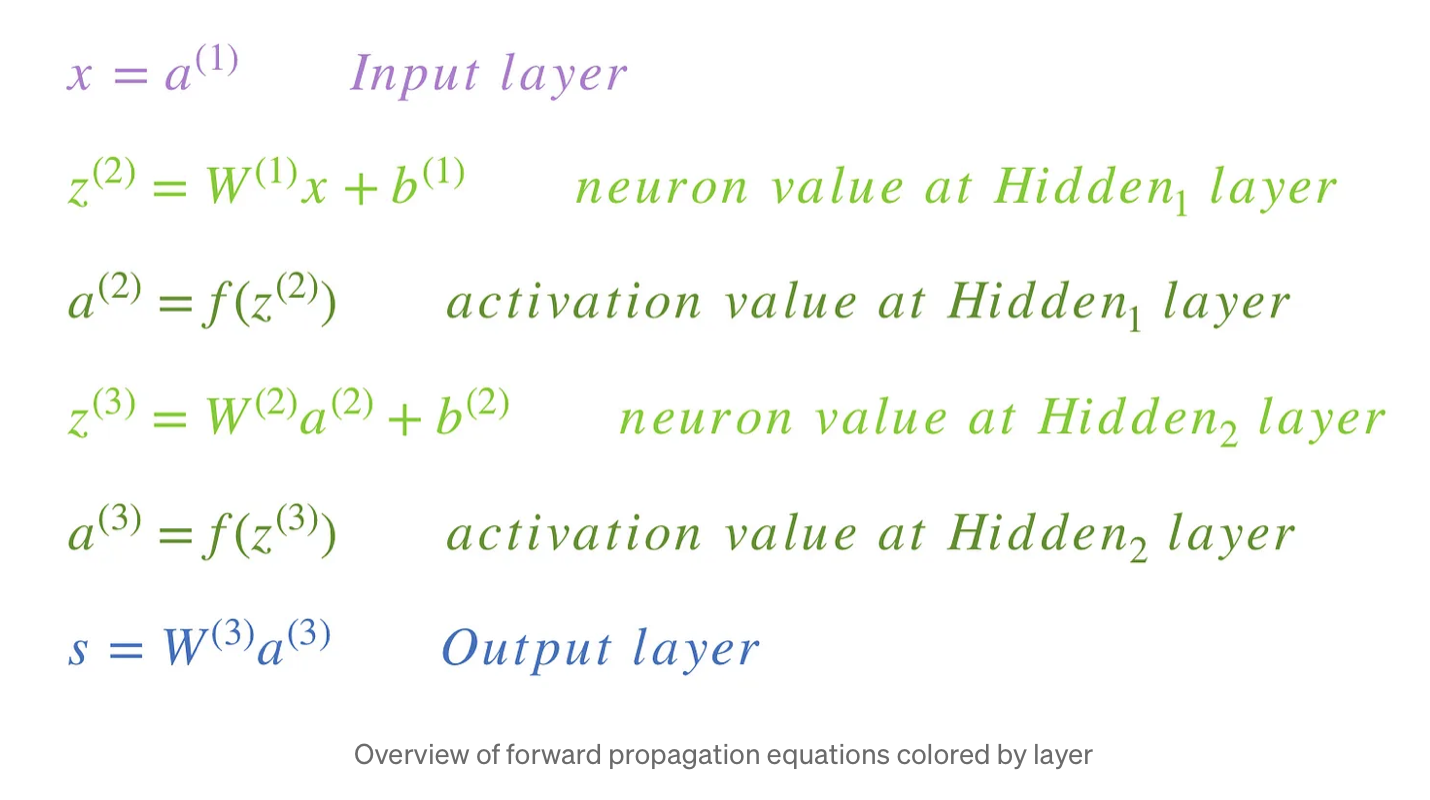
Using matrix representations the input, bias(n=2) and equation for z² is-



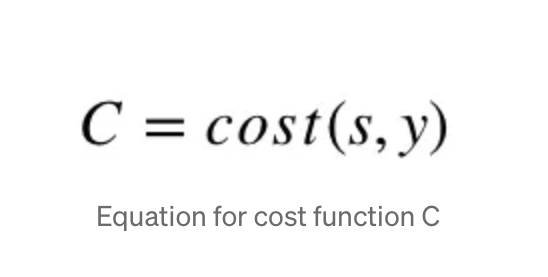
Equation for z² given in matrix manner.

**Output layer-**

**Forward propagation of above network**

****

**The last step is to compare predicted output of s to real output as y(training component of set (x,y))**

Comparison between s and y is drawn by COST function

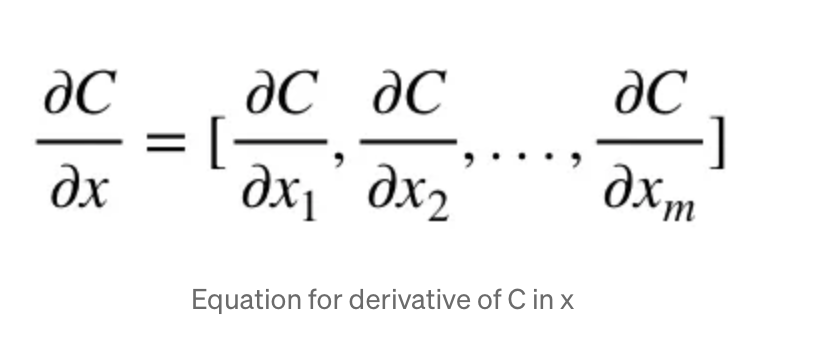
Here cost = MSE

By using C model adjusts the values using blackpropagation to get closer to real output y.

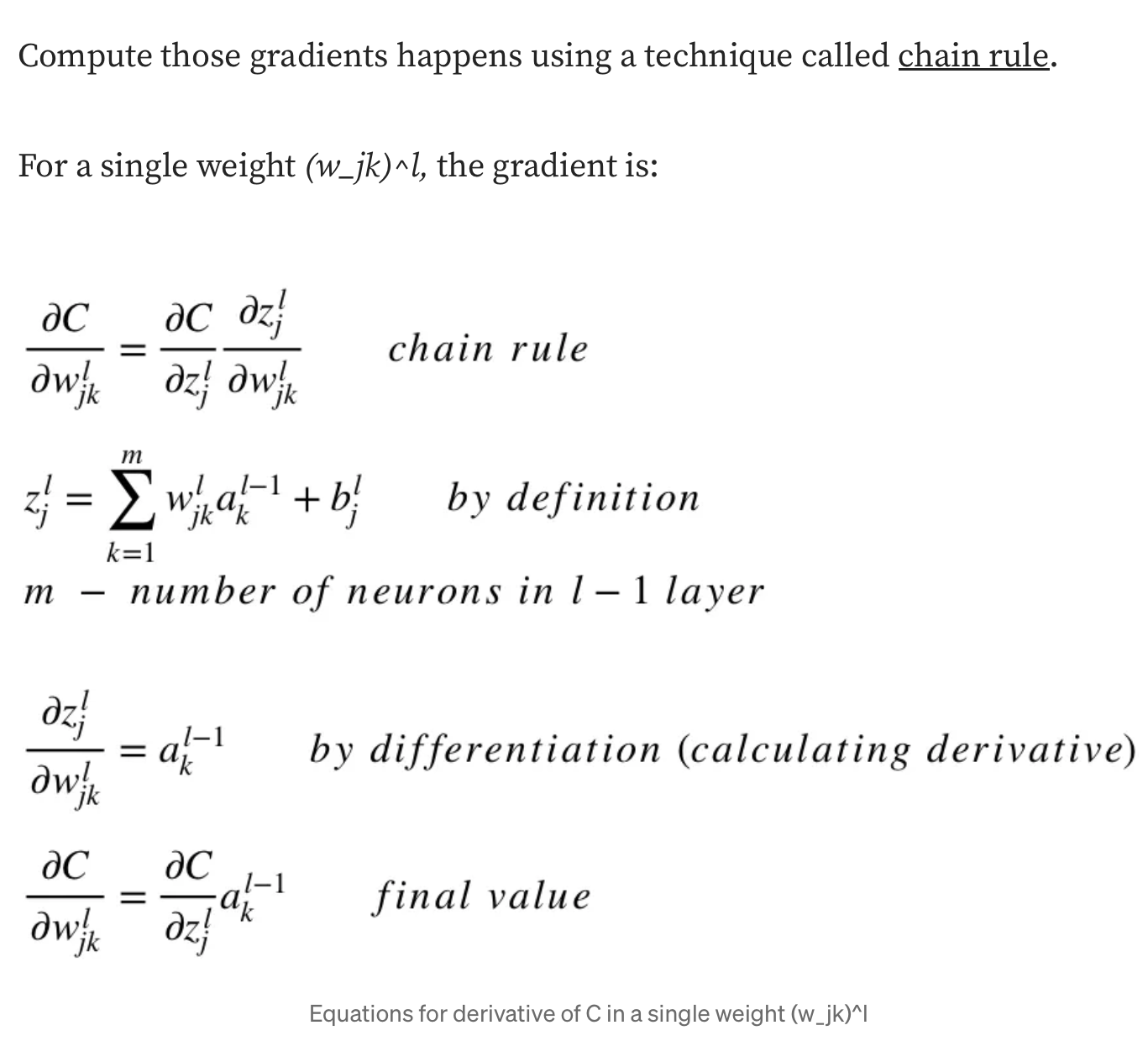
Backpropagation aims to minimize the cost function by adjusting network’s weights and biases.

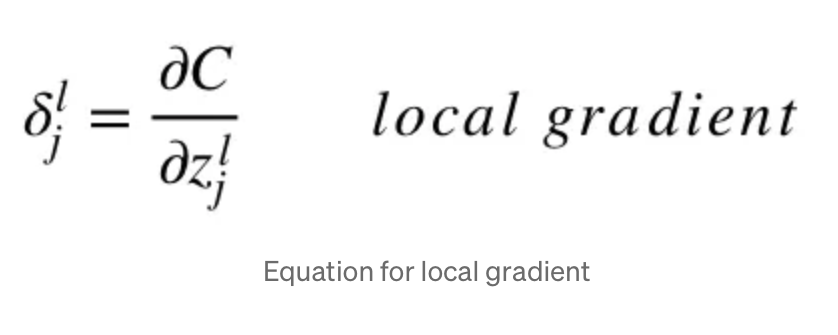
**Gradients-**

Gradient of a function C(x\_1, x\_2, …, x\_m) in point x is a vector of the [partial derivatives](https://en.wikipedia.org/wiki/Partial_derivative) of C in x.

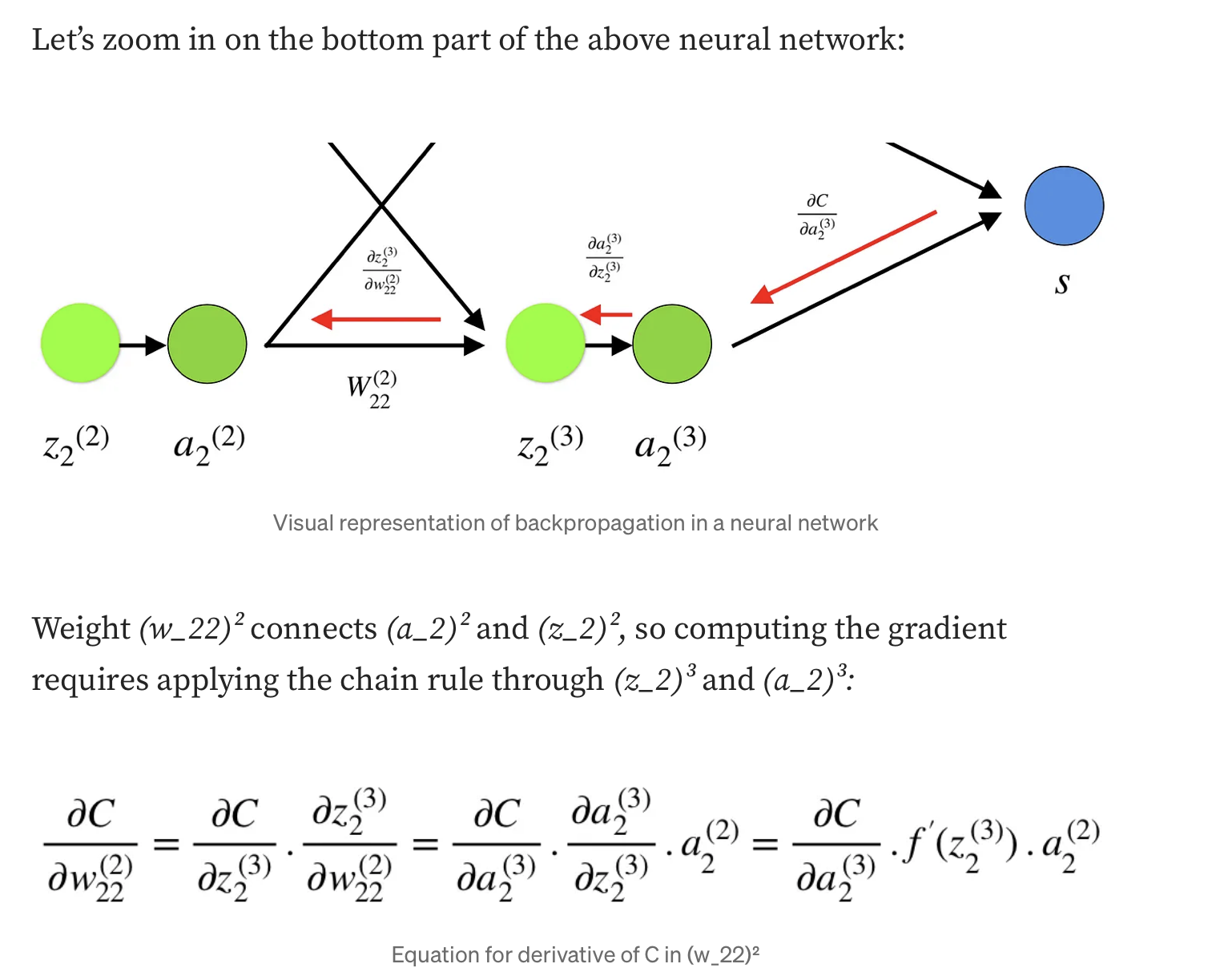


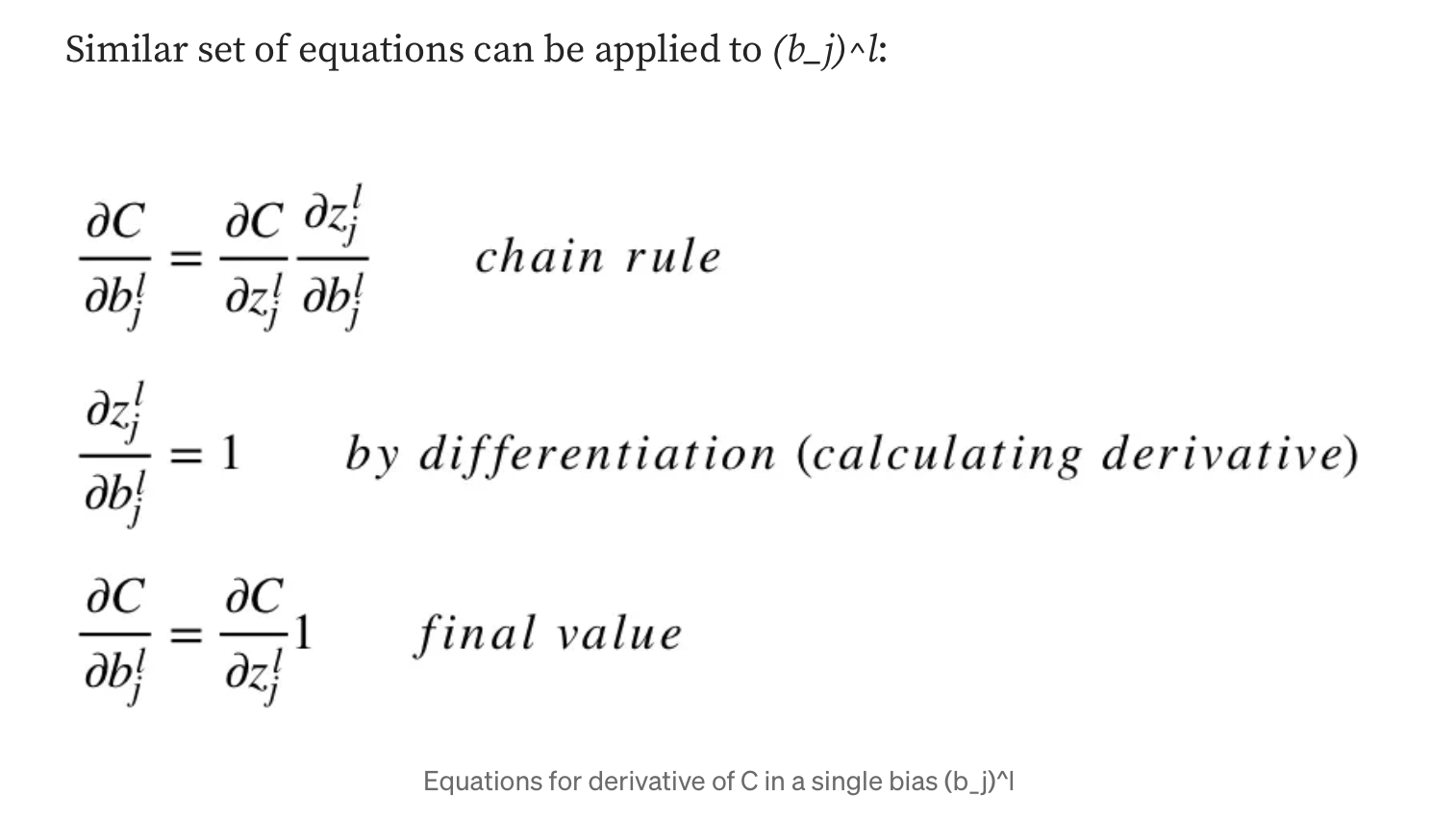
The gradient shows how much the parameter x needs to change (in positive or negative direction) to minimize C.

Common part of above 2 equation is local gradient



**Example-**calculate the gradient of C with respect to a single weight (w\_22)².



BIAS-VARIANCE

\*What is Bias?

The inability of a machine learning algorithm (like linear

regression) to capture true relationship is called bias.

\*What is Variance?

In machine learning lingo, the difference in fits between data sets is called variance.

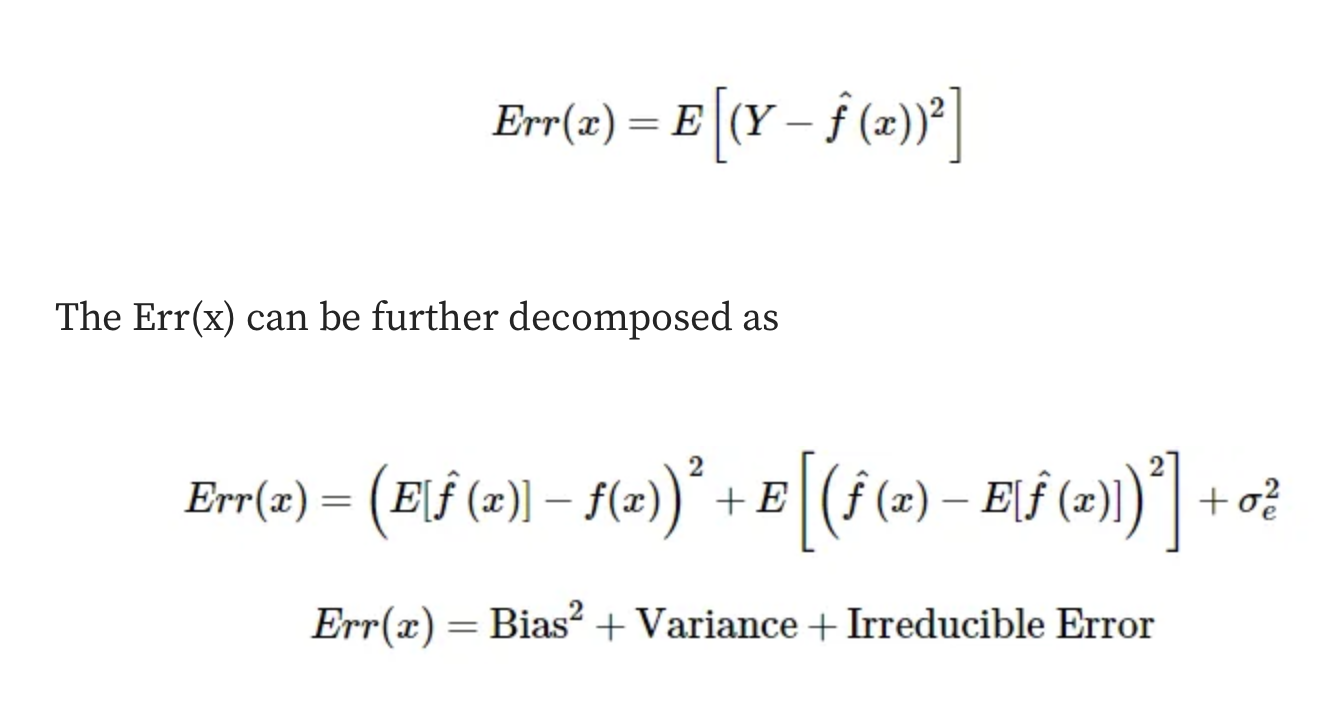
\*Math

We aim to predict Y using covariates X, assuming a relationship expressed as-

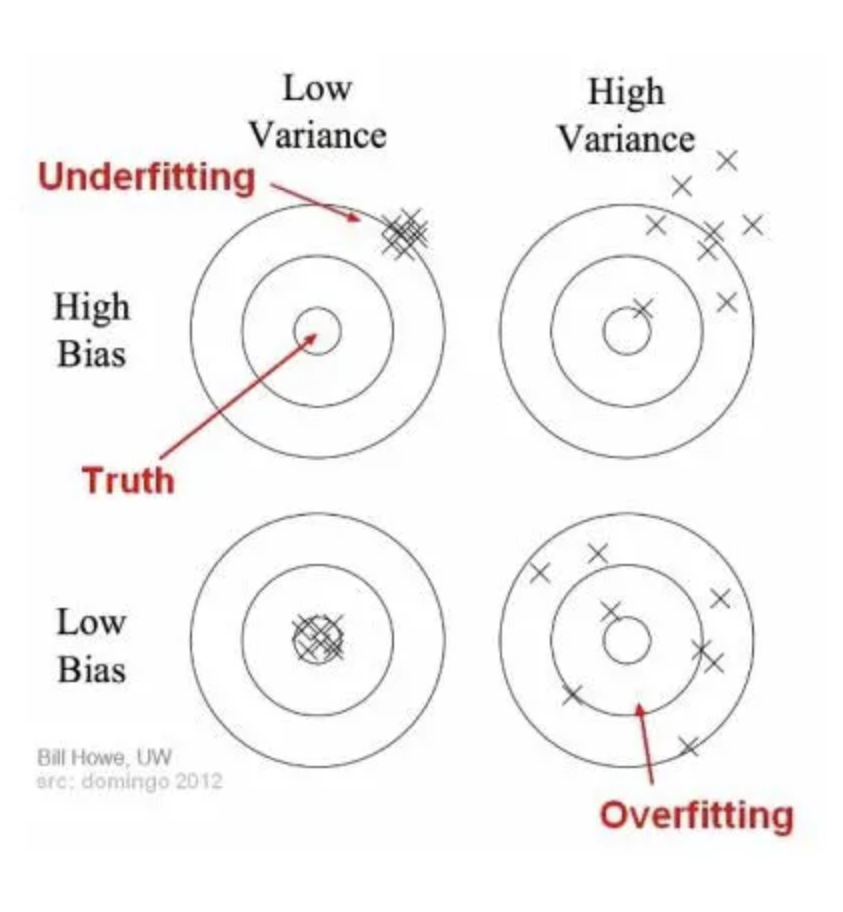
Y=f(X) + e

Here, e = error term(normally distributed with mean =0)

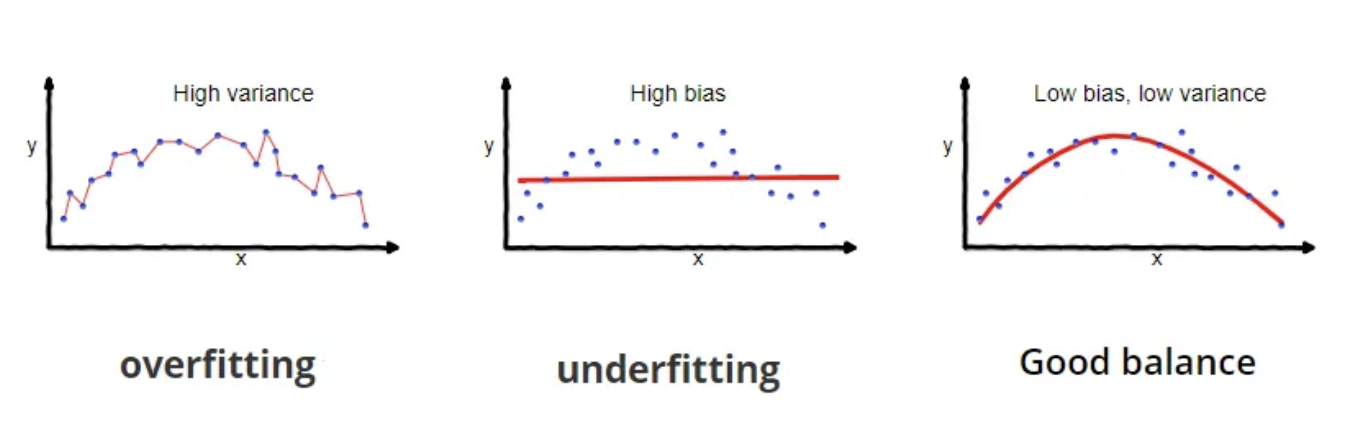
We will make a model f^(X) of f(X) using linear regression -

expected squared error at a point x —>

Irreducible error is the error that can’t be reduced by creating good models. It is a measure of the amount of noise in our data.

BIAS And VARIANCE using bulls-eye representation-

In the diagram, the center of the target signifies a perfect model, while predictions worsen away from it. Underfitting occurs with simplistic models like Linear regression, failing to capture data patterns due to low complexity. Overfitting arises from overly complex models like Decision trees, capturing noise and showing high variance.



\*Why is Bias Variance Tradeoff?

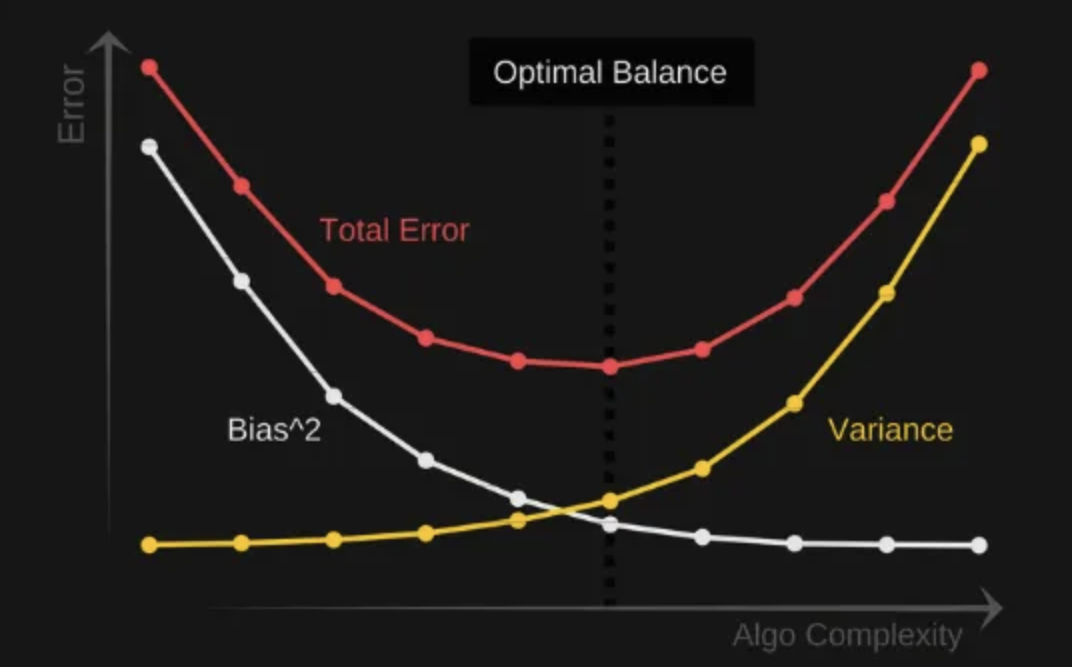
A simple model with few parameters tends to have high bias and low variance, while a complex model with many parameters

exhibits high variance and low bias. Striking a balance without underfitting or overfitting is crucial.

The tradeoff between bias and variance stems from the

complexity dilemma—an algorithm cannot simultaneously be more complex and less complex.

**Total error -**



An optimal balance of bias and variance would never overfit or underfit the model.

\*Regularisation

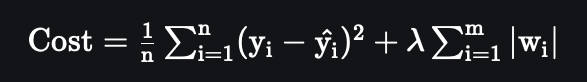
Regularization is a technique used to reduce errors by fitting the function appropriately on the given training set and avoiding overfitting.

Types-

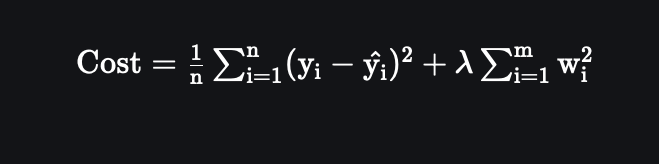
**1.Lasso Regression**

LASSO (Least Absolute Shrinkage and Selection Operator)

regression is a type of regression model that employs L1

Regularization. It adds the absolute magnitude of coefficients as a penalty term to the loss function, aiding in feature selection by pushing insignificant feature weights close to zero.

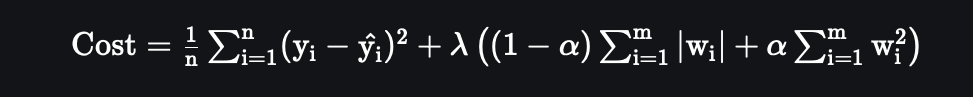
**2.Ridge Regression**

A regression model that uses the L2 regularization technique is called Ridge regression. Ridge regression adds the “squared magnitude” of the coefficient as a penalty term to the loss function(L).

**3.Elastic Net Regression**

This model is a combination of L1 as well as L2 regularization. That implies that we add the absolute norm of the weights as well as the squared measure of the weights. With the help of an extra hyperparameter that controls the ratio of the L1 and L2

regularization.



\*Optimisation

Optimizers are algorithms or methods used to change the attributes of your neural network such as weights and learning rate in order to reduce the losses.

Different types of optimizers-

**1.Gradient Descent**

Gradient Descent is a widely utilized optimization algorithm,

particularly in linear regression, classification, and

backpropagation for neural networks. It relies on first-order

derivatives of a loss function to determine how weights should be adjusted to reach a minimum. Backpropagation transfers loss

between layers, adjusting model parameters (weights) based on these losses to minimize overall loss.

algorithm: θ=θ−α⋅∇J(θ)

**2.Stochastic Gradient Descent**

It’s a variant of Gradient Descent.It tries to update the model’s parameters more frequently.For ex- if dataset contain 1000 rows SGD will update parameters 1000 times in one cycle .

θ=θ−α⋅∇J(θ;x(i);y(i)) , where {x(i) ,y(i)} are the training examples.

**3.Mini-Batch Gradient Descent**

Its best among all variation of gradient descent.It divides data into batches and model updates parameters after every batch.

θ=θ−α⋅∇J(θ; B(i)), where {B(i)} are the batches of training

examples.

\*Momentum

Momentum, introduced to mitigate high variance in SGD, smooths convergence by accelerating towards relevant directions and dampening fluctuations in irrelevant ones.(Reduces the oscillations and high variance of the parameters.)

hyperparameter(γ) is used here.

V(t)=γV(t−1)+α.∇J(θ)

Now, the weights are updated by θ=θ−V(t).

**\*Nesterov Accelerated Gradient**

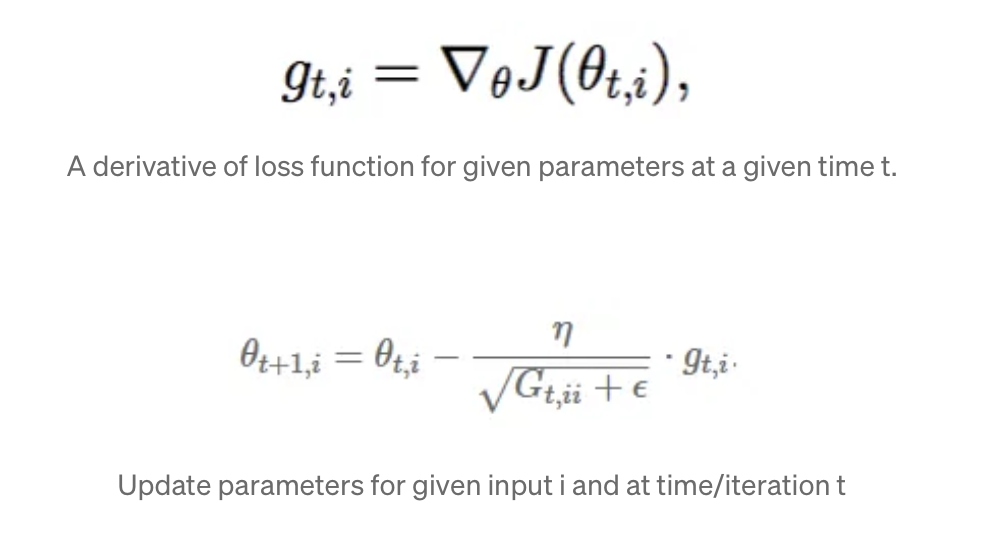
If the momentum is too high the algorithm may miss local minima and continue to rise up.So, to resolve this thing, we use NAG.

We’ll be using γV(t−1) for modifying the weights so, θ−γV(t−1) approximately tells us the future location.Now calculate the cost based on this new parameter.

V(t)=γV(t−1)+α. ∇J( θ−γV(t−1) ) and then update the parameters using θ=θ−V(t).

\*Adagrad

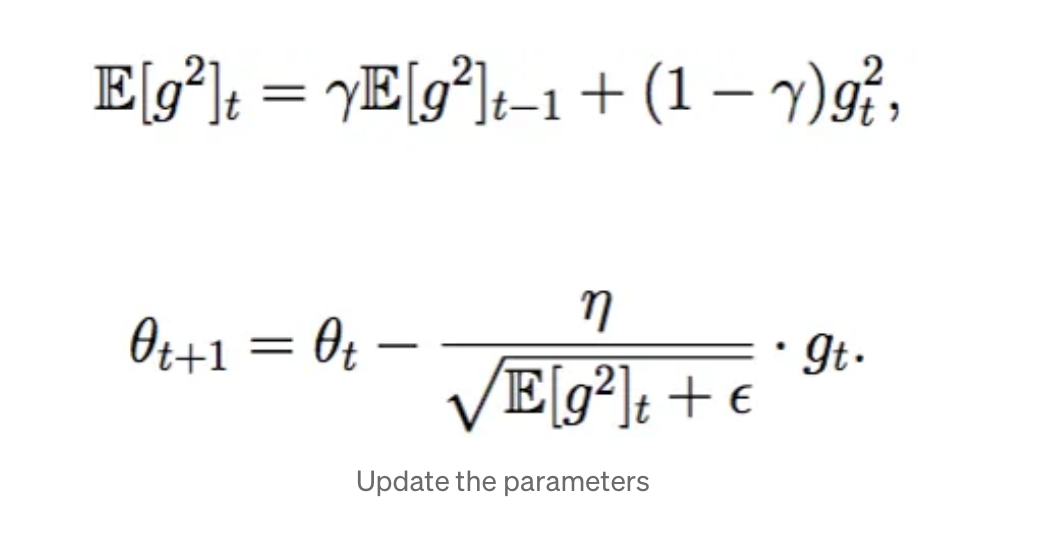
This optimizer addresses a drawback in other optimizers by adjusting the learning rate 'η' individually for each parameter and at every time step 't'. It's a second-order optimization algorithm that operates on the derivative of an error function.



The learning rate η adjusts for each parameter θ(i) based on past gradients, using a smoothing term ϵ to prevent division by zero (usually 1e-8). Interestingly, omitting the square root operation markedly decreases performance.

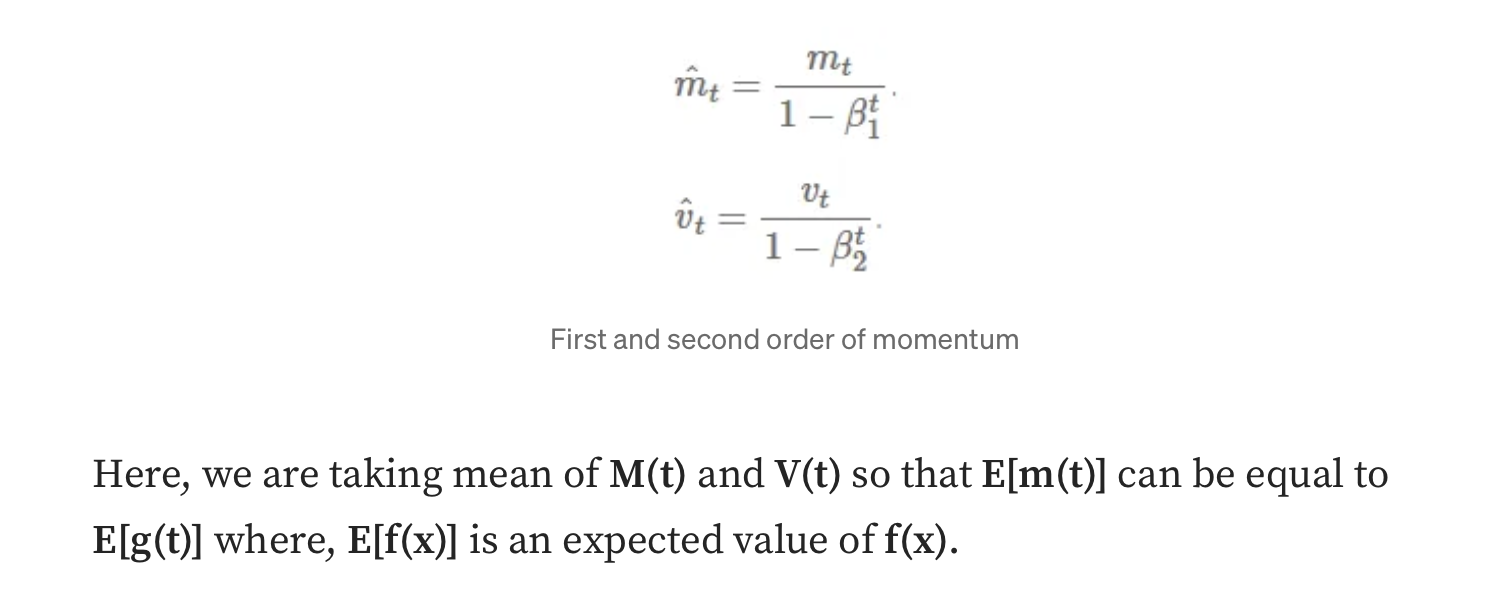
\*AdaDelta

Adadelta extends AdaGrad by addressing its decaying learning rate issue. It limits the accumulation of past squared gradients to a fixed window size 'w', using an exponentially moving average instead of summing all gradients.



\*Adam

Adam (Adaptive Moment Estimation) combines first and second-order momentums for optimization. It moderates velocity to ensure a meticulous search for the minimum. Alongside an exponentially decaying average of past squared gradients, it tracks mean (M(t)) and uncentered variance (V(t)) of gradients as well.



HYPERPARAMETER TUNING

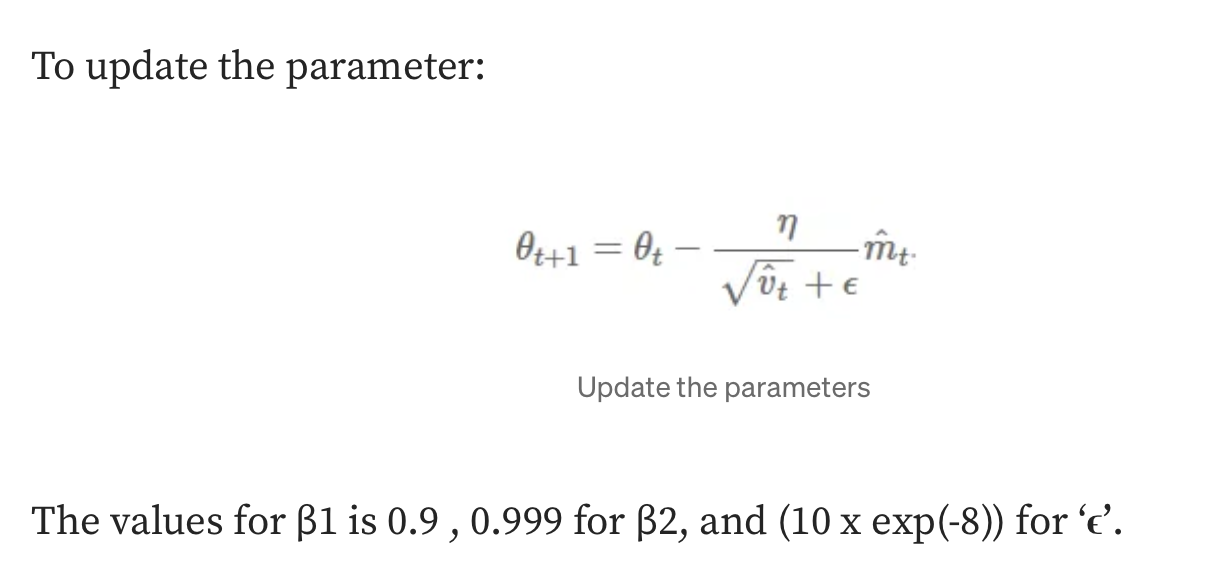
Hyperparameter tuning is the process of selecting the optimal values for a machine learning model’s hyperparameters.

Hyperparameters are settings that control the learning process of the model, such as the learning rate, the number of neurons in a neural network, or the kernel size in a support vector machine. The goal of hyperparameter tuning is to find the values that lead to the best performance on a given task.

**Hyperparamters in neural networks-**

Neural networks have several essential hyperparameters that need to be adjusted, including:

**1.Learning rate:** This hyperparameter controls the step size

taken by the optimizer during each iteration of training. Too small a learning rate can result in slow convergence, while too large a learning rate can lead to instability and divergence.

**2.Epochs:** This hyperparameter represents the number of times the entire training dataset is passed through the model during training. Increasing the number of epochs can improve the

model’s performance but may lead to overfitting if not done carefully.

**3.Number of layers:** This hyperparameter determines the depth of the model, which can have a significant impact on its

complexity and learning ability.

**4.Number of nodes per layer:** This hyperparameter determines the width of the model, influencing its capacity to represent

complex relationships in the data.

**5.Architecture:** This hyperparameter determines the overall structure of the neural network, including the number of layers, the number of neurons per layer, and the connections between layers. The optimal architecture depends on the complexity of the task and the size of the dataset

**6.Activation function:** This hyperparameter introduces

non-linearity into the model, allowing it to learn complex decision boundaries. Common activation functions include sigmoid, tanh, and Rectified Linear Unit (ReLU).

**Hyperparameters in Support Vector Machine-**

**\*Kernel-**

The kernel function that defines the similarity between data points. Different kernels can capture different relationships

between data points, and the choice of kernel can significantly impact the performance of the SVM. Common kernels include linear, polynomial, radial basis function (RBF), and sigmoid.

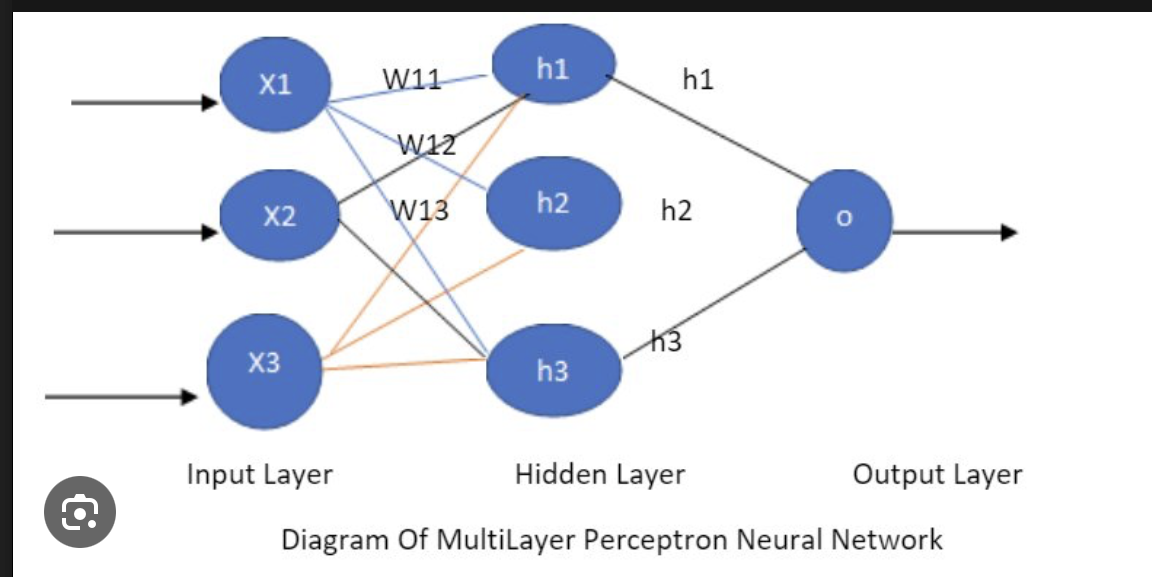
REVIEW QUESTIONS

1.What is Multi-Layer Perceptron?

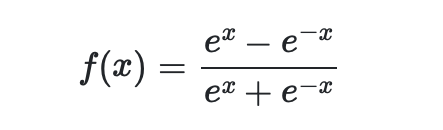
A Multi-Layer Perceptron (MLP) is a type of artificial neural network characterized by multiple layers of interconnected neurons. It consists of an input layer, one or more hidden layers, and an output layer. Each layer except the input layer has a set of

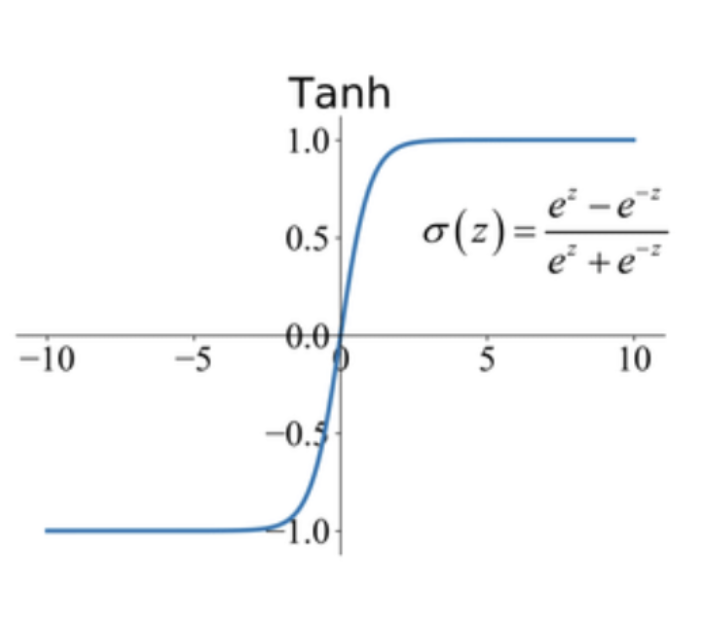
neurons that process input data using activation functions,

enabling complex nonlinear mappings between input and output.



2.What is tanh ?

Tanh Activation is an activation function used for neural networks:

Historically, the tanh function became preferred over the [sigmoid function](https://paperswithcode.com/method/sigmoid-activation) as it gave better performance for multi-layer neural networks.

3.What is inductive Bias and list its advantages in NN?

In NN inductive bias refers to set of assumptions that we take when implementing the model.

Eg,in linear regression we assume variable y is dependent on variable x and in logistic we assume hyperplane

**4.How would the eigenvectors and eigenvalues change during dimensionality reduction?**

The eigenvectors won’t change and will form the basis of lower-dimensional space and as for eigenvalues their magnitude will remain same but their number will decrease.

**5.Difference between Umap and PCA**

**Linearity:** PCA is a linear dimensionality reduction technique, while t-SNE and UMAP are nonlinear techniques. PCA finds

orthogonal axes to explain variance, suitable for global patterns. In contrast, t-SNE and UMAP preserve both local and global structures, ideal for visualizing complex relationships.

**Preservation of Structure:** PCA emphasizes global structure, capturing overall trends but potentially losing local patterns.

t-SNE and UMAP excel at preserving local structure, revealing clusters and nonlinear relationships that might be hidden in high dimensions.

**Computation Time and Scalability:** PCA is computationally

efficient and scalable for large datasets due to its linear algebra approach. In contrast, t-SNE and UMAP are more

computationally intensive, with UMAP designed to be more scalable than t-SNE for handling large-scale datasets efficiently.