**CONCEPTS OF DEEP LEARNING**

**ANN:** Artificial neural network.

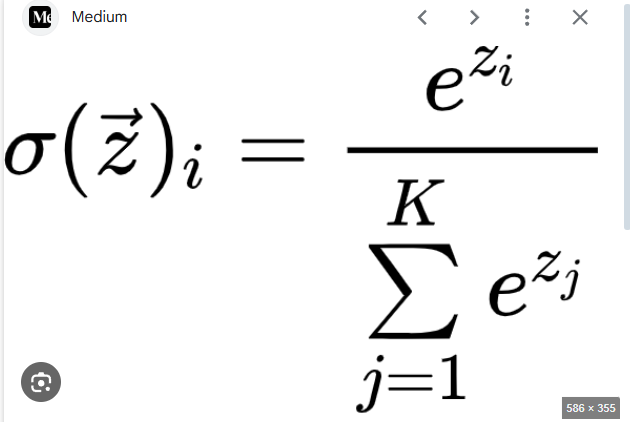
**CNN:** Convolutional neural network.

**RNN:** Recurrent neural network.

Forward propagation:

Backward propagation: When Y raises error, error modifies the weight of functions until it reaches the first nodes.

Softmax function: Solution is always 1.



Monotonic function: x goes up y goes up, vice versa.

Log is important because it can separate small numbers better, and probabilities are very small.

**Entropy**

Can be log base 2 or log base e

Log2 = units: bits

Ln = units: nats

Cross-entropy: Relationship between two probability distributions.

**Derivate**

Derivative is 0 at local minima and maxima points.

Df<0 to the left df>0 right minima

Df>0 left, df<0 right maxima

Vanishing gradient: no minima or maxima but derivative is 0 regardless.

Product rule:

Chain rule:

**Gradient Descent**

DL: 1- Guess, 2-Compute Error, 3-Learn from mistakes and modify parameters.

GD: A way of representing errors in deep learning.

Gradient descent can get stuck if there are multiple local minima of a function. However, in higher dimensional functions (400k+ parameters) it is difficult to have many local minima. Because local minima has to be the same in every direction (every axis) of this function. Probability of +400k parameters in different directions to have the local minima at the same point is lower than a 2 parameter polynomial function.

What to do when GD get stuck anyway?

* If model performs well, don’t worry about it
* Retrain model with new random starting weights, different starting locations for GD.
* Increase model complexity (dimensionality) to have fewer local minima.