⬛ Regularization:

➢ Regularization in deep learning is a method to prevent overfitting by adding constraints or penalties to the neural network's training process.

⬛ Why regularization?

—> to fix overfitting

⬛ Methods of regularization:

- Constraining a model to simplify it
  (Fewer degrees of freedom)
- L1 / L2 regularization (Lasso & Ridge)
- Drop out
- Early Stopping
- Adding more information
- Data augmentation

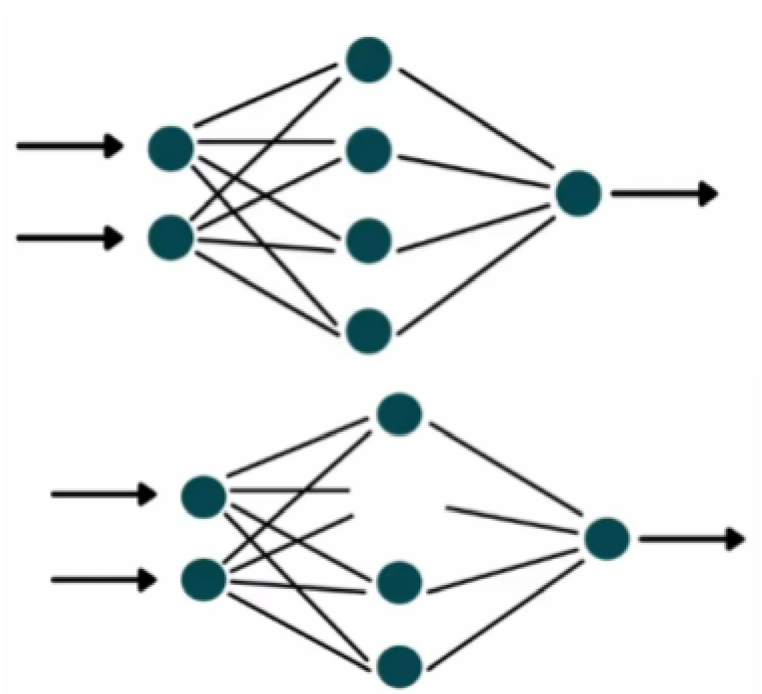
1)L1 & L2 regularization (Lasso & Ridge):

➢ A way to keep the weights low.
➢ **Weight** —> the importance an input has.
It is a parameter associated with a feature or neuron in a model.
➢ L1 and L2 are ways to keep weights low.
➢ Add the weights in loss calculation, effectively penalizing the model for high weight.

◾ **L1 regularization (L1 norm or Lasso):**

➢ Makes the network sparse.
(Yesle chai model ko weights (parameters) lai nearly zero banauna encourage garxa)

➢ Adds the sum of the absolute values of the weights to the loss function.
(Loss function maa penalty term add garxa (i.e $\frac{\lambda \, or \, \alpha}{2n} \sum_{i=1}^{k} W_i$), jasle garxa kunai weight nearly equal to zero banauxe i.e kunai kunai weight lai eliminate garxa.)



◾ L2 Regularization (ridge regression / weight decay):

➢ It is more nuanced.
(Nuanced vannale broad overview matra nabhai, yesle more minute details haru herxa)

➢ Adds sum of the squared values of the weights to the loss.
(Loss function maa penalty term add garxa (i.e $\frac{\lambda \; or \; \alpha}{2n} \sum_{i=1}^{k} (W_i)^2$))

➢ **$\lambda \; or \; \alpha$ = 0(no penalty) to 1(full penalty)**
➢ n = Number of rows
➢ And 2 is used for mathematical convenience .

(Badhi weight bhako node / neuron alik badhi priority maa rakharera penalize garxa ,resulting to less sparse weights.)

◾ The alpha / lambda parameter:

➢ The "alpha" parameter in regularization controls the strength of the penalty applied to model weights.
(Alpha parameter le chai kunai network lai kati panalize garni vanni control dinxa)

➢ Using a **very high/strong alpha** in regularization leads to heavy penalization of model weights, often resulting in a simpler model with smaller weights, which can potentially underfit the data.

(alpha parameter dherai thulo vayo vane, model underfit huna ni sakxa)

➢ Using a **small or too weak alpha** in regularization has minimal impact on the model's weights, potentially allowing them to become too large and leading to overfitting on the training data.
(Overfitting problem solve garna regularization use garinxa ,tara small / too weak alpha / lambda value rakhda tyo overfitting problem nai solve hudaina)

◾ Tips on how to use L1 and L2 regularization:

- Sabai network type maa use hunxan.
- Best result ko lagi input lai normalize garna parxa.
- We can use L1+L2 together
- But L2 is mostly used

2)Dropout Regularization:

➢ Make some neurons inactive.

➢ In every training step ,a neuron has probability p of being inactive.
➢ p = dropout rate
(if p = 0.25, then 25 % neurons dropout hunxan during training time.)
Yesle 2% samma accuracy badhauna sakxa.
➢ p = 0.25, then keep probability = (1-p) = 0.75

➢ The process of dropout regularization in deep learning involves the following steps:

1. **Initialization:**

   Start by initializing your neural network with all its neurons and connections.

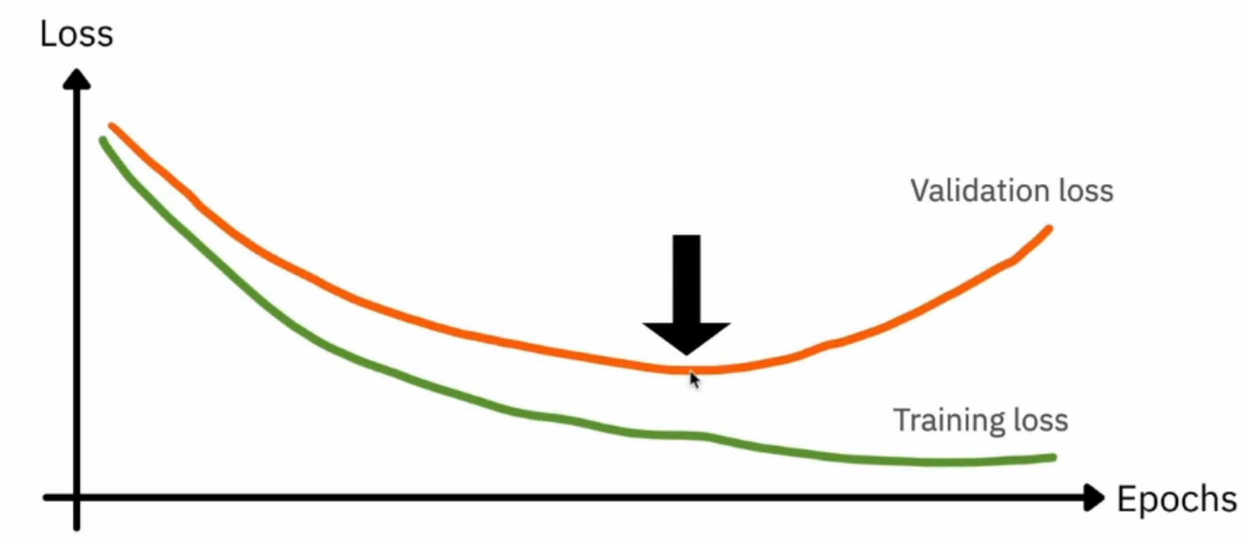
2. **Training Phase:**
   - During training, for each mini-batch of data:
   - Apply dropout: Randomly select a subset of neurons to be "dropped out" (deactivated). The probability of keeping each neuron active is determined by the "keep probability," usually set between 0.5 and 1.
   - Forward Pass: Perform a forward pass through the network, where the dropped-out neurons have no impact on the current calculation.

    - Backward Pass: Compute the gradients and update the weights of the active neurons based on the loss for the current mini-batch.
    - Repeat these steps for multiple epochs and mini-batches.
3. **Inference Phase:**
    - During inference or when making predictions, dropout is turned off, and all neurons are active.
    - The predictions are made using the complete network, without any dropout.

3)Early stopping:

➢ Early stopping is a regularization technique where the training process is halted when the model's performance on a validation dataset starts to degrade, preventing overfitting.

➢ Stops the learning process once the validation error is minimum.



(validation loss minimum vayeko point patta lagayera tei nera training lai halt garxa)

4)Data Augmentation:

➢ Data augmentation is a technique to enhance a machine learning model's performance by artificially expanding the training dataset through various transformations of the original data, reducing overfitting.

➢ Create variations of the data points to enrich the training data .