# Chapter 8 (10)
# Innleiðing í tilgjørdum viti

Notes

## Contents

# 1 Reinforcement Learning

**Definition:** Reinforcement learning (RL) is a type of machine learning where an agent learns to make decisions by interacting with an environment. The agent receives rewards or punishments based on the actions it takes, and learns to maximize cumulative rewards over time.

**Basic Loop:**

- Agent observes the current **state** of the environment.
- Agent chooses an **action**.
- Environment returns a new **state** and a **reward**.
- The agent updates its policy based on the experience.


### RLHF: Reinforcement Learning from Human Feedback

Used for training large language models. The steps:

1. Prompt is given to the model.
2. Model generates multiple responses.
3. Human annotators rank the responses.
4. A reward model is trained to predict human preferences.
5. The main model is updated using reinforcement learning to improve future responses.

# 2 Unsupervised Learning

**Definition:** Learning patterns in data without labeled outcomes.

### Clustering

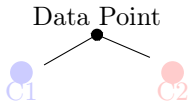
Grouping objects such that similar ones are in the same cluster.

**Applications:**

- Genetic research
- Image segmentation
- Market research
- Medical imaging
- Social network analysis

### K-Means Clustering

1. Initialize $k$ cluster centers randomly.
2. Assign each point to the nearest cluster center.
3. Recalculate the cluster centers.
4. Repeat until convergence.




# 3 Neural Networks

## Biological Neurons

Neurons receive electrical impulses from others, process them, and may activate (fire). Artificial Neural Networks (ANNs) are inspired by this concept.

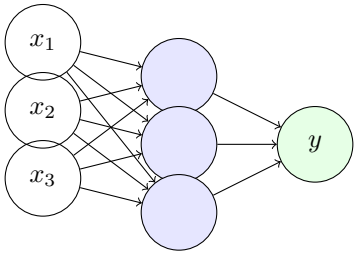
## Artificial Neural Network (ANN)

A neural network is a parameterized function that maps input features $x_1, \ldots, x_n$ to an output $y$:

$$y = F(x_1, \ldots, x_n)$$

**Key Properties:**

- **Input layer:** Accepts features.
- **Hidden layers:** Process and transform input.
- **Output layer:** Produces prediction.




**Activation Function: ReLU**

$$\text{ReLU}(x) = \max(0, x)$$

# 4 Training Neural Networks

## Gradient Descent

An optimization algorithm used to minimize the loss function by updating weights in the opposite direction of the gradient.

- Initialize weights randomly.
- Compute gradient of loss with respect to weights.
- Update weights:
$$w := w - \eta \cdot \nabla L(w)$$
where $\eta$ is the learning rate.

## Mini-batch Gradient Descent

Instead of using the entire dataset, a small random batch is used to compute the gradient.

## Backpropagation

Used to train networks with hidden layers.

1. Forward pass to compute output.
2. Compute loss (error).
3. Backward pass to propagate error and compute gradients.
4. Update weights using gradient descent.

# 5 Dense Neural Networks

- Each neuron is connected to every neuron in the previous and next layer.
- Can learn linear patterns.
- Limit: Only learns linearly separable functions.

**Solution:** Use multiple layers (Multilayer Perceptrons) to learn complex patterns.

**Depth:** Number of layers.
**Width:** Number of neurons per layer.
**Size:** Maximum width of all layers.

# 6 Training Strategies

## Train/Validation Split

- 80% for training
- 20% for validation

**Overfitting:** Model memorizes training data and fails to generalize.
**Underfitting:** Model is too simple or lacks data.
**Solution:**

- Get more data or augment existing data.
- Tune hyperparameters (e.g., number of layers, learning rate).

## Dropout Regularization

Dropout randomly disables a fraction of neurons during training to prevent overfitting.

- Drop rate = proportion of units set to zero.
- During inference, all units are used, but activations are scaled by $\frac{1}{1-\text{rate}}$.