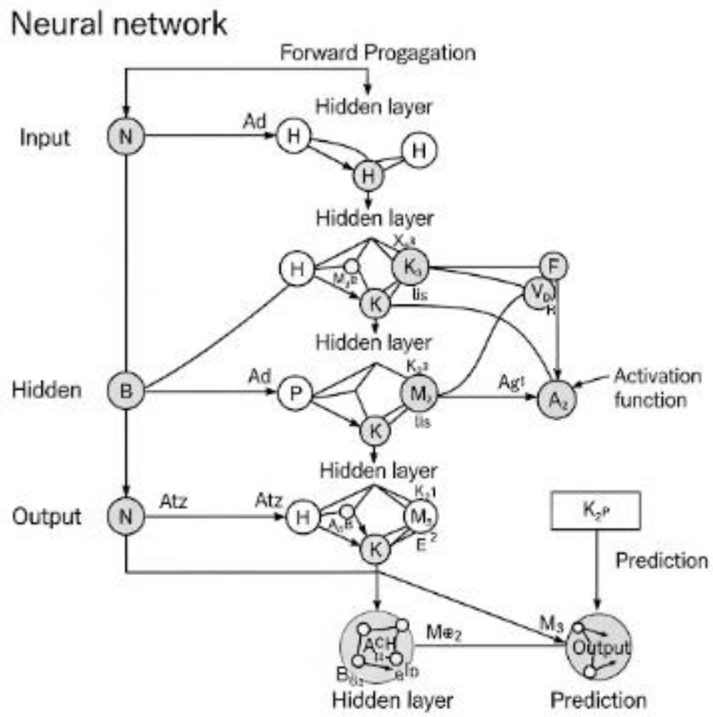
Visualizing Neural Network Components

Subtitle: Using TensorFlow Playground for Model Training

Mohammad Anwar Karim

05/25/2025

Neural Network Structure



- The graphic demonstrates data flow from the input layer through numerous hidden levels to the output layer. Each circle symbolizes a neuron, while lines denote weights.

Layers

- The graphic demonstrates data flow from the input layer through numerous hidden levels to the output layer. Each circle symbolizes a neuron, while lines denote weights.
- Layers organize neurons in a neural network.

- **Input Layer:** Receives data.
- **Hidden Layers:** Process input using weights and activation functions.
- **Output Layer:** Produces the final prediction.
📌 *Each layer transforms data progressively for better feature extraction.*

Neurons

- Neurons are the basic units of a neural network. They receive inputs, apply weights, compute sums, and pass the results through an activation function to generate outputs.

Neurons (nodes) are the basic units of computation.

- They sum weighted inputs and apply an activation function.
  📌 *the circles labeled H, K, M, etc., represent neurons.*

Weights

- Weights are parameters that determine the influence of input data on a neuron's output. Training adjusts weights to minimize the loss function and improve predictions.

Weights are parameters that scale input values.

- Lines connecting neurons carry weights.
- During training, weights are adjusted to minimize error. 📌 *Stronger weights lead to a stronger influence on the output.*

Activation Functions

- Activation functions add non-linearity to the model. Common types include ReLU, Sigmoid, and Tanh. They help the network learn complex patterns and relationships in the data.

Activation functions introduce non-linearity.

- Examples: Sigmoid, ReLU, Tanh
- Help networks learn complex patterns
  📌 *this is shown between nodes and output transformation.*

Loss Functions

- Loss functions measure the error between predicted and actual outputs. Common examples are Mean Squared Error and Cross Entropy Loss. Lower loss means better model accuracy.

A loss function calculates the error between predicted and actual outputs.

- Guides learning by quantifying how “wrong” predictions are.
- Common: Mean Squared Error, Cross-Entropy
  📌 *This guides the optimization step.*

Optimization Algorithms

- Optimization algorithms such as Gradient Descent and Adam adjust weights during training to minimize the loss function, improving model performance over iterations.

These update weights to reduce the loss.

- Use **gradient descent** or variants like Adam or SGD.
- Learn by iteratively improving predictions.
📌 *The model learns through backpropagation based on loss gradients.*

Summary & Insights

- Adding more hidden layers increases model complexity but may lead to overfitting.
- Activation functions are crucial for learning non-linear relationships.
- Visualizing training helps understand the balance between **underfitting** and **overfitting.**
- Hyperparameters like learning rate and batch size significantly affect performance.
- 🎯 **Conclusion:**
  Visualizing neural networks clarifies how data transforms through layers. It also highlights the importance of tuning architecture and parameters for optimal learning.