Neural Networks with PyTorch - Tutorial

Your Name

September 29, 2025

Outline

What are Neural Networks?

- Computational models inspired by biological neural networks
- Consist of interconnected nodes (neurons) in layers
- Learn complex patterns from data

Key Components

- Neurons/Nodes: Basic processing units
- Weights: Strength of connections
- Biases: Help fit the data
- Activation Functions: Introduce non-linearity
- Layers: Input, hidden, output

PyTorch Fundamentals

- Dynamic computational graphs
- Automatic differentiation (autograd)
- GPU acceleration
- Rich ecosystem

Tensors in PyTorch

- Multi-dimensional arrays (like NumPy)
- Support for GPU and autograd
- Basic operations: creation, addition, multiplication, reshaping

Automatic Differentiation

- PyTorch computes gradients automatically
- Essential for backpropagation and training
- Example: $y = x^2 + 3x + 1$

Neural Network Building Blocks

- Activation functions: ReLU, Sigmoid, Tanh, Leaky ReLU
- Linear layers (fully connected)
- Dropout for regularization
- Batch normalization for stability

Example: ImprovedNet

- Multiple linear layers
- Dropout and batch normalization
- Forward pass: Linear $\rightarrow$ BatchNorm $\rightarrow$ ReLU $\rightarrow$ Dropout

Training a Neural Network

- Forward pass: compute predictions
- Loss calculation: e.g., MSELoss for regression
- Backward pass: compute gradients
- Optimizer step: update parameters

Evaluation and Metrics

- Plot training/test loss curves
- Use metrics: MSE, R^2, accuracy
- Visualize predictions vs. true values

Best Practices and Tips

- Start simple, increase complexity gradually
- Use dropout and batch normalization
- Monitor training and validation performance
- Use appropriate loss functions and optimizers

Further Resources

- PyTorch Documentation: `https://pytorch.org/docs/`
- PyTorch Tutorials: `https://pytorch.org/tutorials/`
- Deep Learning Book: `https://www.deeplearningbook.org/`
- Papers With Code: `https://paperswithcode.com/`

Questions?