Lesson 36: Overfitting Prevention

Data Science with Python – BSc Course

45 Minutes

Learning Objectives

The Problem: Neural networks have millions of parameters and can memorize training data. How do we ensure they generalize to new data?

After this lesson, you will be able to:

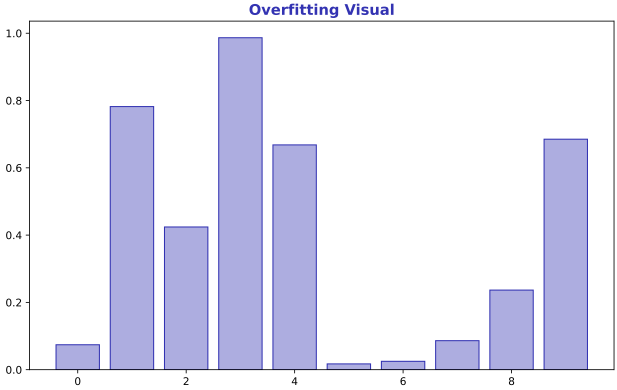
- Apply dropout regularization
- Use early stopping to prevent overfitting
- Diagnose overfitting from learning curves
- Build robust neural network models

Finance Application: Preventing models from fitting to noise in market data

Recognizing Overfitting

The Gap Between Train and Test

- Training loss keeps decreasing, validation loss increases
- Model memorizes training data instead of learning patterns

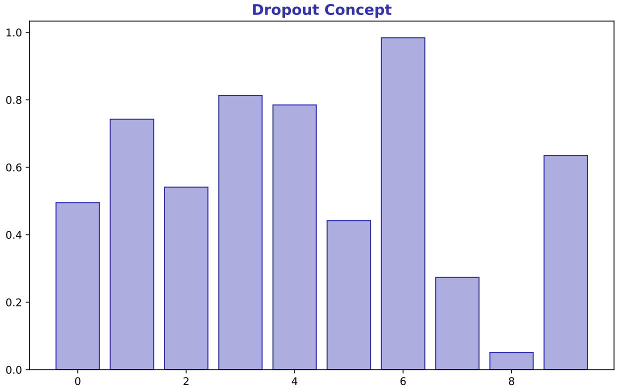


Sign: train accuracy 99%, test accuracy 60% = severe overfitting

Dropout Concept

Randomly Ignoring Neurons

- During training: randomly set fraction of neurons to zero
- Forces network to not rely on any single neuron

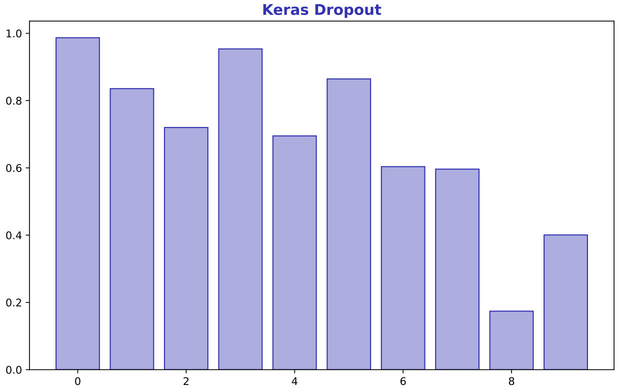


Dropout = training many thinned networks, averaging at test time

Keras Dropout Layer

Implementation

- `model.add(Dropout(0.5))` – drop 50% of neurons
- Place after Dense layers, typically 0.2-0.5 rate

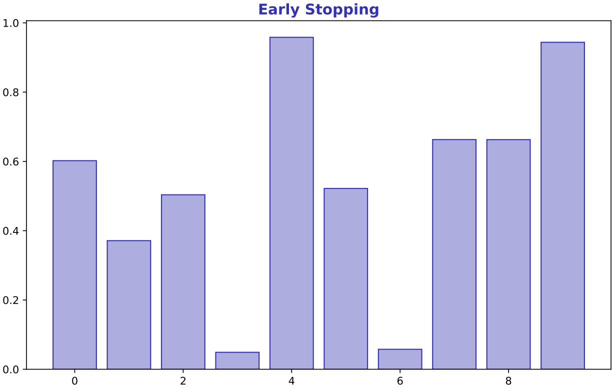


Common pattern: Dense → Dropout → Dense → Dropout

Early Stopping

Stop Before Overfitting

- Monitor validation loss; stop when it starts increasing
- `EarlyStopping(patience=10, restore_best_weights=True)`

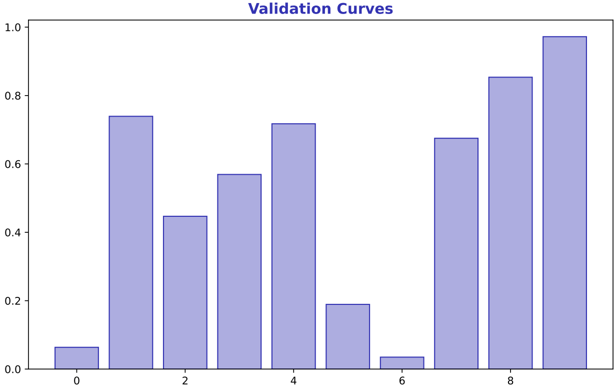


Patience = how many epochs without improvement before stopping

Validation Curves

Reading the Learning Curve

- Underfitting: both train and val loss high
- Overfitting: train low, val high (growing gap)
- Good fit: both low, small gap

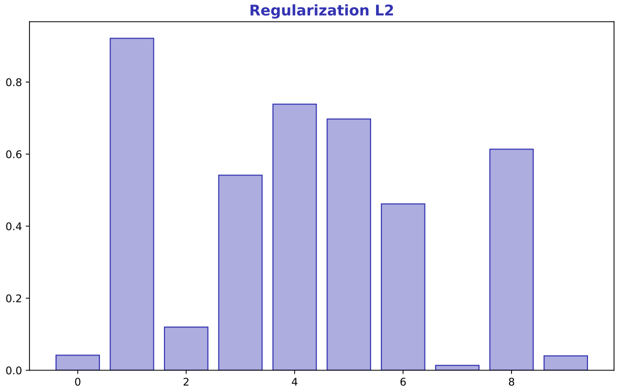


Always plot train and val loss together during development

L2 Regularization

Penalizing Large Weights

- Add $\lambda \sum w^2$ to loss function
- Keras: `Dense(64, kernel_regularizer=l2(0.01))`

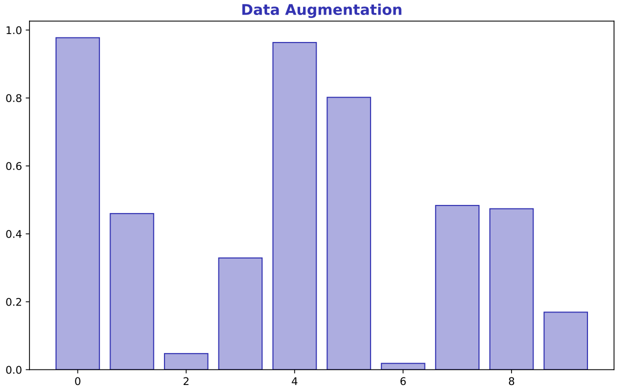


L2 regularization keeps weights small, reducing model complexity

Data Augmentation

Creating More Training Data

- For images: rotate, flip, crop, adjust brightness
- For time series: add noise, time warping

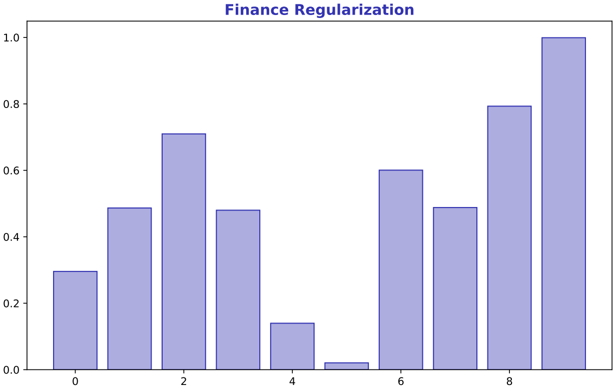


More diverse training data = better generalization

Finance Regularization

Special Considerations for Financial Data

- Financial data is noisy – regularization is essential
- Use dropout + early stopping + small networks



Rule: simpler models often work better on financial data

Hands-On Exercise (25 min)

Task: Compare Regularization Techniques

1. Create overfit-prone dataset (few samples, many features)
2. Train baseline MLP – observe severe overfitting
3. Add Dropout(0.5) – compare train/val curves
4. Add EarlyStopping – when does training stop?
5. Compare final test accuracy across all variants

Deliverable: Side-by-side learning curves + accuracy comparison table.

Extension: Try combining dropout + L2 + early stopping

Lesson Summary

Problem Solved: We can now prevent neural networks from memorizing data and ensure generalization.

Key Takeaways:

- Dropout: randomly zero neurons during training
- Early stopping: halt when validation loss stops improving
- L2 regularization: penalize large weights
- Always monitor train vs validation loss curves

Next Lesson: Text Preprocessing (L37) – preparing text for NLP

Memory: Dropout = random zeros. Early stopping = stop at best val. Watch the gap.