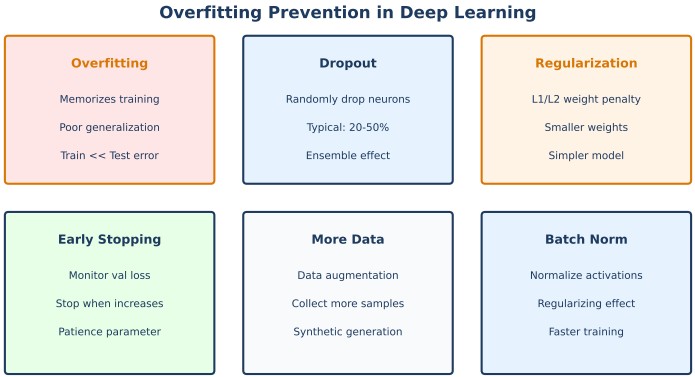
# Lesson 36 Summary: Overfitting Prevention

## Data Science with Python – Key Concepts

Data Science Program

# Overfitting Prevention Overview




**Multiple techniques prevent neural networks from memorizing data**

# The Overfitting Problem

**Why it happens:**

- **Too many parameters:** Model capacity exceeds data
- **Training too long:** Memorizes noise
- **Symptom:** Low train error, high test error

**Overfitting = poor generalization to new data**

# Dropout

**Random neuron deactivation:**

- **Training:** Randomly zero neurons (20-50%)
- **Inference:** Use all neurons, scale outputs
- **Effect:** Prevents co-adaptation

**Dropout creates implicit ensemble of networks**

# L1 and L2 Regularization

**Weight penalty in loss:**

- **L2 (Ridge):** $\lambda \sum w_i^2$ shrinks weights
- **L1 (Lasso):** $\lambda \sum |w_i|$ creates sparsity
- **Effect:** Simpler, smoother models

**L2 is more common in deep learning (weight decay)**

# Early Stopping

**Stop before overfitting:**

- **Monitor:** Validation loss each epoch
- **Stop when:** Val loss increases (patience)
- **Restore:** Best weights from training

**Simple yet effective regularization technique**

# Batch Normalization

**Normalize layer inputs:**

- **Normalize:** Zero mean, unit variance per batch
- **Learn:** Scale and shift parameters
- **Bonus:** Regularizing effect, faster training

---

**BatchNorm placed after linear, before activation**

# Data Augmentation

**Increase effective dataset size:**

- **Images:** Rotate, flip, crop, color shift
- **Time series:** Add noise, time warping
- **Finance:** Bootstrap, synthetic scenarios

**More data is the best regularizer**

# Best Practices

**Combine multiple techniques:**

- **Start simple:** Small network first
- **Add capacity:** If underfitting
- **Add regularization:** If overfitting

**Balance model capacity with regularization strength**

# Quick Reference

**Regularization Techniques:**

| Technique | How It Works |
|---|---|
| Dropout | Randomly zero neurons during training |
| L2/Weight decay | Penalize large weights |
| Early stopping | Stop when val loss increases |
| Batch norm | Normalize layer inputs |
| Data augmentation | Artificially expand dataset |

**Use multiple techniques together for best results**