Deep Learning Cheat Sheet

A concise reference for core deep learning concepts, especially for building and evaluating baseline models.

---

1. Activation Functions

Function	Description
ReLU	Outputs `max(0, x)`
Sigmoid	Squashes input to [0, 1]
Tanh	Zero-centered sigmoid
GELU	Smooth ReLU used in BERT

Common Activation Function Formulas

Sigmoid:

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

Tanh:

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

---

2. Training Parameters

- **Batch Size**: Number of samples per training step
- **Epoch**: One full pass through the training set
- **Learning Rate**: Step size for parameter updates
- **Dropout**: Randomly disables neurons to reduce overfitting
- **Weight Decay**: L2 regularization to penalize large weights

---

3. Normalization Techniques

Batch Normalization

- Normalizes activations across batch dimension
- Helps stabilize training and speeds up convergence

Layer Normalization

- Normalizes across features of each individual sample
- Common in sequence models like Transformers

---

4. Bias and Variance

- **Bias**: Error due to overly simple models (underfitting)
- **Variance**: Error from sensitivity to training data (overfitting)

Bias-Variance Tradeoff Table

Model Type	Bias	Variance	Risk
Underfit	High	Low	Poor capacity to learn
Overfit	Low	High	Poor generalization
Balanced	Low	Low	Ideal

---

5. Evaluation Metrics

- **Accuracy** = correct predictions / total samples
- **Macro F1 Score** = average F1 across all classes (equal weight)
- Use **macro F1** when class distribution is balanced

---

6. Attention Mechanism (Transformers)

Formula:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Where:

- (Q) = query
- (K) = key
- (V) = value

Intuition:

- **Query** = what we're looking for
- **Key** = what each token offers
- **Value** = information that will be passed forward if attended to

---

7. Softmax

- Converts logits into a probability distribution

Formula:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

- Outputs are between 0 and 1
- Sum of outputs = 1

---

8. Why Non-Linearity Matters

Linear models can't capture complex decision boundaries.
Without non-linear activation functions, stacked layers collapse into a single linear transformation.

Even simple non-linear functions (e.g., ReLU) break that limitation.

---

9. Overfitting vs. Underfitting

Signs of Overfitting

- Training loss decreases
- Validation loss increases
- High variance (model too flexible)

Signs of Underfitting

- Training loss is high
- Model cannot learn patterns
- High bias (model too simple)

---

10. Regularization Techniques

- **Dropout**: Deactivates a percentage of neurons per forward pass
- **Weight Decay**: Penalizes large weights (L2 regularization)
- **Early Stopping**: Stops training when validation loss plateaus

---

11. Optimization Strategy (Hyperparameter Tuning)

Recommended Tuning Order:

1. Learning rate
2. Dropout
3. Hidden layer size

4. Batch size
5. Weight decay
6. Embedding dimension
7. Optimizer
8. Activation function

Tune 1–2 hyperparameters at a time to avoid noise and overfitting to validation.

References

- Vaswani et al., "Attention Is All You Need" (2017)
- Stanford CS231n Notes: https://cs231n.github.io/
- FastAI Deep Learning Book: https://book.fast.ai