LECTURE 7 CHEATSHEET

# Diagnose before you regularize

Use training data to fit. Use validation data to choose. Open the test set once.

**Regularization supplies a preference where the training data are silent. Validation chooses its strength.**

## 1 · Keep the split contract clean

| | |
|---|---|
| TRAIN | Fit parameters $\theta$. |
| VALIDATE | Choose the recipe, checkpoint, and every knob. |
| TEST | Estimate final performance once, after choices are frozen. |

## 2 · Read the curves before choosing a cure

| Pattern | Diagnosis | First move |
|---|---|---|
| train low; val high | high variance | regularize or add valid data |
| both high | high bias | improve model, features, or optimization |
| val rises late | overtraining | restore the validation-best checkpoint |
| gap small; both poor | still underfit | do not regularize harder |

### Bias-variance lens

$$E\left[\left(\hat{f}(x) - f(x)\right)^2\right]$$

$$= \underbrace{\left(E\left[\hat{f}(x)\right] - f(x)\right)^2}_{\text{bias}^2} + \underbrace{\text{Var}\left(\hat{f}(x)\right)}_{\text{variance}} + \underbrace{\sigma^2}_{\text{noise}}$$

Regularization usually trades a little more **bias** for less **variance**. Validation tells you whether the trade helped.

## 3 · One dataset, four outcomes

| Degree-9 fit | train MSE | validation MSE |
|---|---|---|
| no regularizer | 0.0000 | **1.70** |
| weight decay | - | 0.065 |
| early stopping | - | 0.061 |
| input jitter | - | **0.048** |

Same ten measurements and same flexible model. Different preferences choose different solutions.

## 4 · Choose the intervention site

| Site | Method | Knob | Preference |
|---|---|---|---|
| objective | weight decay | $\lambda$ | smaller weights |
| loop | early stopping | $t^\star$ | simpler training trajectory |
| data | augmentation / noise | strength | valid nearby views |
| model | dropout | $p_{\text{drop}}$ | redundant features |
| targets | label smoothing | $\varepsilon$ | finite confidence |

Mixup and CutMix act on both the examples and their targets. Change one site at a time when you can.

## 5 · Weight decay: prefer smaller weights

**Intuition:** among equally good fits, prefer the one that does not need large coefficients.

$$\mathcal{J}(w) = \frac{1}{2n}\|Xw - y\|^2 + \frac{\lambda}{2}\|w\|^2$$

$$w_{t+1} = \underbrace{(1 - \eta\lambda)w_t}_{\text{shrink}} \underbrace{- \eta g_t}_{\text{data step}}$$

Sweep $\lambda$ on a log scale. Too little leaves a high-variance fit; too much underfits.

Optional detail: with adaptive optimizers, decoupled AdamW keeps the shrink separate from the adaptive gradient step.

## 6 · Early stopping: select the best checkpoint

**Intuition:** later updates can fit training quirks after the useful pattern has already been learned.

$$t^\star = \underset{t}{\text{argmin}}\, \mathcal{L}_{\text{val}(w_t)}$$

1. save when validation improves
2. stop after patience is exhausted
3. restore the saved best state

The last epoch is not automatically the selected model.

**Practical order: diagnose, pick one site, sweep its knob on validation, and keep the simplest recipe that actually helps.**

LECTURE 7 CHEATSHEET

# Operate each regularizer correctly

The mechanism matters: what changes during training, what stays fixed, and which knob validation chooses.

## 1 · Augmentation: teach valid variation

**Intuition:** show task-valid versions of each sample so the model cannot rely on one stored form.

$$A_{\tau(x,y)} = \left(g_{\tau(x)}, h_{\tau(y)}\right)$$

For every transform, decide:

- **KEEP** the label is unchanged
- **UPDATE** move boxes, masks, keypoints, spans, or times
- **REJECT** the transformed example is no longer valid

Input jitter reuses the target locally:

$$\varepsilon \sim \mathcal{N}(0, \sigma^2 I), \quad (x_i, y_i) \to (x_i + \varepsilon, y_i)$$

$\sigma$ sets the neighbourhood size. Train transforms are stochastic; validation and test stay deterministic. Audit real batches before training.

## 2 · Mixup and CutMix: constrain the space between samples

**Why it can help:** the prediction must change smoothly between examples instead of forming a sharp, memorized boundary around each one.

$$\lambda \sim \mathrm{Beta}(\alpha, \alpha), \quad \tilde{x} = \lambda x_i + (1-\lambda) x_j,$$

$$\tilde{y} = \lambda y_i + (1-\lambda) y_j$$

A 70/30 cat-dog blend contributes $-0.70 \log p_{\text{cat}} - 0.30 \log p_{\text{dog}}$.

**Before using it:** ask whether the mixed input makes sense, whether the mixed target says what you want, and whether untouched validation data improves.

$\alpha$ controls how strongly examples mix; $\lambda$ is redrawn. CutMix uses the pasted area fraction for the soft target.

## 3 · Dropout: practise with missing features

**Intuition:** random feature outages during training reward backup routes, so one feature cannot depend on one fixed partner.

Let $q_{\text{keep}} = 1 - p_{\text{drop}}$:

$$m_i \sim \mathrm{Bernoulli}(q_{\text{keep}}), \quad \tilde{h}_i = \frac{m_i h_i}{q_{\text{keep}}}$$

$$E\left[\tilde{h}_i\right] = h_i, \quad \frac{\partial \mathcal{L}}{\partial h_i} = \left(\frac{m_i}{q_{\text{keep}}}\right) \frac{\partial \mathcal{L}}{\partial \tilde{h}_i}$$

- **TRAIN** fresh mask; keep survivors scaled by $\frac{1}{q_{\text{keep}}}$
- **EVAL** all units on; no mask and no extra scaling

Backward reuses the forward mask: a kept path is scaled by $\frac{1}{q_{\text{keep}}}$; a dropped path gets zero gradient for that pass. PyTorch scales automatically. `eval()` changes module behaviour; `inference_mode()` disables autograd.

## 4 · Label smoothing: ask for confidence, not certainty

**Intuition:** one-hot cross-entropy keeps rewarding a larger correct-class logit gap. Smoothing gives that push a finite stopping point.

$$\tilde{y} = (1-\varepsilon) y + \frac{\varepsilon}{K} \mathbf{1}, \quad \frac{\partial \mathcal{L}}{\partial z} = p - \tilde{y}$$

With $K = 5$ and $\varepsilon = 0.1$:

$$(0, 1, 0, 0, 0) \to (0.02, 0.92, 0.02, 0.02, 0.02)$$

The gradient stops pushing class $b$ when $p_b = 0.92$. Against any wrong class $j$, the finite target gap is

$$m^\star = z_b - z_j = \ln\left(\frac{0.92}{0.02}\right) \approx 3.83.$$

$m^\star$ is the desired correct-vs-wrong logit margin. Validate $\varepsilon$: smoothing can soften too much and can erase useful class-similarity structure.

## 5 · A disciplined recipe

1. fit a reproducible baseline and plot learning curves
2. add only task-valid augmentation
3. sweep weight decay on a log scale
4. add one targeted method only if the curves justify it
5. freeze the recipe and checkpoint; evaluate test once

Keep the training budget and validation protocol fixed during comparisons.

## 6 · Final audit

- ✓ Learning curves diagnose the problem.
- ✓ One intervention site and one named knob.
- ✓ Validation chose the recipe and checkpoint.
- ✓ The best checkpoint was restored.
- ✓ The test set remained untouched until the end.
- ✓ Seed, splits, schedule, and every knob were logged.

**If you have not plotted learning curves, diagnose the problem before choosing a regularizer.**

### One-line map

**Weight decay** constrains parameters.
**Early stopping** constrains training time.
**Augmentation** expands valid training views.
**Mixup/CutMix** constrain behaviour between examples.
**Dropout** disrupts hidden feature pathways.
**Label smoothing** softens the confidence target.