LECTURE 10 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 θ
VALIDATE	choose method, strength, schedule, and checkpoint
TEST	estimate final performance once choices are frozen

2 · Read the curves first

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 validation-best checkpoint
gap small; both poor	underfit	do not regularize harder

Bias–variance lens

$$\text{prediction error} = \text{bias}^2 + \text{variance} + \text{noise}$$

$$\text{bias} = E\big[\hat{f}(x)\big] - f(x), \quad \text{variance} = \text{Var}\big(\hat{f}(x)\big).$$

Regularization often accepts a little more bias to reduce variance. Validation tests whether that trade improves unseen data.

3 · Compare at fixed conditions

Keep the split, model family, optimizer budget, augmentation policy, and evaluation pipeline fixed while changing one regularization knob.

Training loss is allowed to rise. The relevant question is whether validation performance and stability improve.

4 · Choose the intervention site

Site	Method	Knob	Preference
objective	weight decay	λ	smaller weights
loop	early stopping	$t^\star$	shorter useful trajectory
data	augmentation	strength	valid nearby views
model	dropout	p_{drop}	redundant features
targets	label smoothing	ε	finite confidence

Mixup and CutMix act on examples and targets. Change one site at a time when possible.

5 · Weight decay

$$\mathcal{J}(w) = \mathcal{L}_{\text{data}(w)} + \frac{\lambda}{2}\|w\|^2$$

For SGD, the update can be read as

$$w_{t+1} = (1 - \eta\lambda)w_t - \eta g_t.$$

Sweep λ on a log scale. Too little leaves high variance; too much underfits.

With adaptive optimizers, AdamW decouples shrinkage from adaptive gradient scaling.

6 · Early stopping

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

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.

Diagnose → choose one site → sweep its knob on validation → keep the simplest recipe that helps.

LECTURE 10 CHEATSHEET

Operate each regularizer correctly

Know what changes during training, what stays fixed, and which knob validation chooses.

1 · Augmentation: valid variation

$$A_{\tau(x,y)} = \big(g_{\tau(x)}, h_{\tau(y)}\big).$$

For every transform, decide:

- **KEEP** the label is unchanged
- **UPDATE** move boxes, masks, keypoints, spans, or times
- **REJECT** the transformed example is not task-valid

Train transforms are stochastic; validation and test transforms are deterministic. Inspect actual batches before training.

2 · Mixup and CutMix

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

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

The model is asked to vary smoothly between examples. Validate α, and ask whether both the mixed input and soft target make sense for the task.

CutMix uses the pasted area fraction for the target weight.

3 · Dropout: missing-feature practice

With keep probability $q = 1 - p_{\text{drop}}$:

$$m_i \sim \mathrm{Bernoulli}(q), \quad \tilde{h}_i = m_i \frac{h_i}{q}.$$

Then $E\big[\tilde{h}_i\big] = h_i$.

- **TRAIN** fresh mask; surviving units scaled by $\frac{1}{q}$
- **EVAL** all units active; no mask or extra scaling

`model.eval()` changes dropout behaviour. `inference_mode()` changes autograd; they solve different problems.

4 · Label smoothing

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

With $K = 5$ and $\varepsilon = 0.1$, the correct-class target is 0.92 and each wrong-class target is 0.02. The gradient stops demanding infinite logit separation.

Validate ε: too much smoothing can erase useful confidence or class-similarity structure.

5 · A disciplined recipe

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

6 · Final audit

- ✓ Learning curves support the diagnosis.
- ✓ One intervention site and one named knob.
- ✓ Validation chose the strength and checkpoint.
- ✓ The best checkpoint was restored.
- ✓ Test stayed untouched until the end.
- ✓ Seeds, splits, schedule, and every knob were logged.

One-line map

Weights constrain parameters.
Stopping constrains training time.
Augmentation expands valid views.
Dropout disrupts feature pathways.
Smoothing softens confidence targets.

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