Neural Networks Mini-Exercise (ReLU by Hand)
Hidden Features, Nonlinearity, and Interactions

How this handout connects to the slides

In the slides, we described a neural network as a model that (i) forms **weighted sums** of inputs, (ii) applies a **nonlinear activation** (here: ReLU), and (iii) combines learned **hidden features** to produce a prediction.

The goal of this in-class exercise: do one complete *forward pass* by hand (no calculus, no training), then translate each neuron into a plain-English rule. This is the core mental model of neural networks.

Toy setting (for easy arithmetic). We use two inputs:

$$X_1 = \text{“Hits score”} \in \{0, 1, 2\}, \qquad X_2 = \text{“Experience score”} \in \{0, 1, 2\}.$$

Think of 0/1/2 as low/medium/high. In real applications, inputs are continuous, but the logic is the same.

Quick toolbox

Activation (ReLU). $\text{ReLU}(z) = \max(0, z)$. If $z < 0$, the neuron outputs **0** (“off”). If $z > 0$, it outputs z (“on”).

Exhibits

Figure 1: Exhibit 1: A tiny $2 \to 2 \to 1$ neural network (ReLU hidden layer)

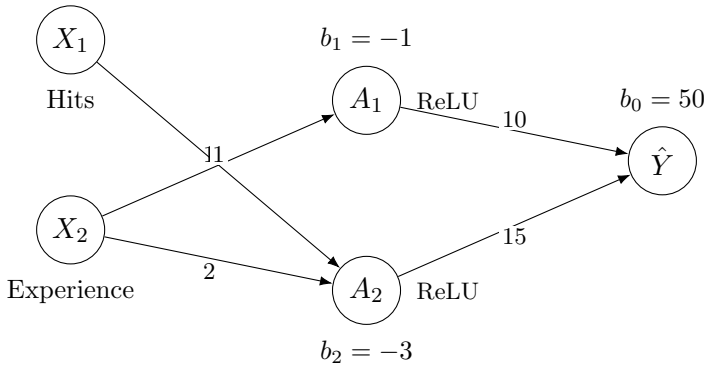

Table 1: Exhibit 2: The model (numbers chosen for hand calculation)

Hidden layer (create two learned features)	**Interpretation (plain English)**
$z_1 = b_1 + 1 \cdot X_2 = -1 + X_2, \quad A_1 = \max(0, z_1)$	Neuron A_1 turns "on" only when experience is high enough to pass a threshold.
$z_2 = b_2 + 1 \cdot X_1 + 2 \cdot X_2 = -3 + X_1 + 2X_2, \quad A_2 = \max(0, z_2)$	Neuron A_2 depends on **both** hits and experience, so it can capture **interactions**.
Output layer (combine learned features)	
$\hat{Y} = b_0 + 10A_1 + 15A_2 = 50 + 10A_1 + 15A_2$	Final prediction is a weighted combination of the learned features.

Table 2: Exhibit 3: Worksheet (fill in by hand)

Case	X_1	X_2	$z_1 = -1 + X_2$	$A_1 = \max(0, z_1)$	$z_2 = -3 + X_1 + 2X_2$	$A_2 = \max(0, z_2)$	$\hat{Y} = 50 + 10A_1 + 15A_2$
1 (Rookie, low hits)	0	0					
2 (Rookie, high hits)	2	0					
3 (Some exp., high hits)	2	1					
4 (Veteran, low hits)	0	2					
5 (Veteran, med hits)	1	2					
6 (Veteran, high hits)	2	2					

In-Class Questions

1. **Forward pass (do the mechanics).** Fill in Exhibit 3. Work left-to-right: compute z_1, z_2, apply ReLU to get A_1, A_2, then compute $\hat{Y}$.

2. **Translate neurons into business rules (no math symbols).** Complete the sentences:

   - Neuron A_1 turns on only when _____.
   - Neuron A_2 turns on only when _____.

   (Hint: "turns on" means the inside score z is positive.)

3. **Where does nonlinearity happen (ReLU as a gate)?** Find a pair of cases in which the same hidden neuron switches from *off* to *on* as inputs change (i.e., z crosses 0). Describe (i) what ReLU does to that neuron's contribution and (ii) how this creates a "kink" / piecewise-linear change in $\hat{Y}$ (hint: Case 2 vs. Case 3).

4. **Interaction (the punchline).** Compare the *impact of hits* for rookies vs. veterans:

   - Rookie: compare Case 1 vs. Case 2 (same experience, hits increase).
   - Veteran: compare Case 4 vs. Case 6 (same experience, hits increase).

   Is the change in $\hat{Y}$ the same or different? Explain in one sentence using the idea of a neuron being "on" or "off."

5. **Knob-turning intuition (how training feels).** Suppose the model is systematically **too low** for high-experience/high-hits people (like Case 6). Which single parameter would you increase first, and why? Choose one: 15 (weight on A_2), 10 (weight on A_1), or 50 (intercept).

6. **Counterfactual check (what if we remove ReLU?).** Imagine we replace ReLU by the identity function ($A = z$). In plain English, what capability would the model lose?

Solution Manual

Model recap

$$z_1 = -1 + X_2, \quad A_1 = \max(0, z_1), \qquad z_2 = -3 + X_1 + 2X_2, \quad A_2 = \max(0, z_2),$$
$$\hat{Y} = 50 + 10A_1 + 15A_2.$$

Interpretation cues:

- A_1 is an **experience threshold feature**: it activates only when experience is sufficiently high.
- A_2 is a **combined performance feature**: it activates only when hits and experience together exceed a threshold, with experience weighted more heavily (coefficient 2 on X_2).

Q1. Forward pass: completed worksheet

Table 3: Completed Exhibit 3 (solutions)

Case	X_1	X_2	z_1	A_1	z_2	A_2	$\hat{Y}$
1 (Rookie, low hits)	0	0	−1	0	−3	0	50
2 (Rookie, high hits)	2	0	−1	0	−1	0	50
3 (Some exp., high hits)	2	1	0	0	1	1	65
4 (Veteran, low hits)	0	2	1	1	1	1	75
5 (Veteran, med hits)	1	2	1	1	2	2	90
6 (Veteran, high hits)	2	2	1	1	3	3	105

Q2. Plain-English neuron rules

- **Neuron A_1 turns on only when experience is very high.** Formally, $A_1 > 0$ when $z_1 = -1 + X_2 > 0 \Rightarrow X_2 > 1$. With $X_2 \in \{0, 1, 2\}$, this means A_1 turns on only at $X_2 = 2$ (veteran).
- **Neuron A_2 turns on only when the combined score of hits and experience is high enough.** Formally, $A_2 > 0$ when $z_2 = -3 + X_1 + 2X_2 > 0 \Rightarrow X_1 + 2X_2 > 3$. In words: high experience can compensate for lower hits (because experience gets double weight), but rookies need unusually high hits to activate the neuron (and in this toy grid they still do not).

Q3. Where nonlinearity happens (what ReLU changes)

ReLU creates **threshold behavior**: a neuron can be completely off ($A = 0$) until its score crosses 0. Two clean examples from the worksheet:

- **Case 2 (rookie, high hits):** $z_2 = -1 < 0$ so $A_2 = 0$ despite relatively high X_1. The network refuses to "credit" hits through this channel until the combined threshold is met.
- **Case 3 (some experience, high hits):** $z_2 = 1 > 0$ so $A_2 = 1$ and the contribution *turns on* immediately, raising $\hat{Y}$ from 50 (Case 2) to 65 (Case 3).

Teaching point: without ReLU, contributions change smoothly and linearly everywhere; with ReLU, the model is **piecewise linear** with **kinks** at activation thresholds.

Q4. Interaction: why hits matter more for veterans here

Compute the changes:

- **Rookies (Case 1 → Case 2):** $X_2 = 0$ in both cases. $A_1 = 0$ and $A_2 = 0$ in both cases, so $\hat{Y}$ stays at 50.
  Interpretation: in the rookie region, the network has both hidden features off, so increasing hits does not move the prediction (in this toy setup).
- **Veterans (Case 4 → Case 6):** $X_2 = 2$ in both cases. $A_1 = 1$ in both cases, but A_2 rises from 1 to 3 as hits rise. $\hat{Y}$ increases from 75 to 105 (a +30 change).
  Interpretation: for veterans, the A_2 channel is on and sensitive to hits, so hits have a large effect.

One-sentence interaction explanation (model language): *The effect of X_1 on $\hat{Y}$ depends on X_2 because X_2 determines whether A_2 activates and how large it becomes.*

Q5. Knob-turning (which parameter to change and why)

Given the problem statement ("too low" specifically for high-experience/high-hits cases like Case 6), the best single adjustment is typically:

- **Increase the weight on A_2 (currently 15).** Reason: A_2 is the feature that becomes large in the high-experience/high-hits region. Increasing its weight raises predictions most where A_2 is active and large, and has little effect where $A_2 = 0$.

Why not the other two (in this scenario)?

- Increasing the intercept 50 shifts *all* cases upward (including rookies), which does not target the specific region.
- Increasing the weight on A_1 (currently 10) increases all veteran cases by the same amount (since $A_1 = 1$ whenever $X_2 = 2$), but it does not specifically address the *high-hits* part of the high-high region.

Optional deeper note (if students ask about thresholds): To make A_2 activate *earlier* (e.g., for $X_2 = 1$ more often), you would adjust the bias $b_2 = -3$ upward (less negative). That is a different kind of "knob" (a threshold knob).

Q6. Counterfactual: what capability is lost if ReLU is removed

If we replace ReLU with the identity ($A = z$), the model becomes purely linear in (X_1, X_2):

$$\hat{Y} = 50 + 10(-1 + X_2) + 15(-3 + X_1 + 2X_2) = -5 + 15X_1 + 40X_2.$$

What is lost:

- **No gating / thresholds:** contributions cannot turn off; they always apply everywhere.
- **No context-dependent effects:** hits always have the same marginal impact (15 per unit) regardless of experience. The "hits matter more for veterans" pattern disappears.

Take-home core ideas

- **Neurons create learned features.** Each hidden neuron computes a weighted score z and then applies an activation to produce a feature (A_1 or A_2) that is *useful for prediction.*
- **Where nonlinearity comes from (ReLU = a "gate").** ReLU turns each neuron into a *piecewise* rule:

$$A_k = \max(0, z_k) = \begin{cases} 0 & \text{if } z_k \leq 0 \quad \text{(off)} \\ z_k & \text{if } z_k > 0 \quad \text{(on)} \end{cases}$$

  This creates **kinks/thresholds** in the model. For example, when $X_2 = 0$ (rookie), $z_1 = -1 + X_2 < 0$, so $A_1 = 0$ and the A_1 channel contributes *nothing.* When $X_2 = 2$ (veteran), $z_1 > 0$, so $A_1 > 0$ and the contribution turns on. The overall prediction $\hat{Y}$ therefore changes **nonlinearly** as inputs cross these thresholds.
- **How interactions are captured (effects depend on context).** Neuron A_2 depends on both inputs: $z_2 = -3 + X_1 + 2X_2$. That means:
  - Whether A_2 is **on or off** depends on the **combination** (X_1, X_2).
  - The **impact of hits** (X_1) on $\hat{Y}$ can change with experience (X_2): when $A_2 = 0$ (off), increasing X_1 may have *no* effect through that channel; when $A_2 > 0$ (on), increasing X_1 raises $\hat{Y}$ via the $15A_2$ term. This is a concrete interaction: "hits matter more when experience is high enough for A_2 to activate."
- **Why more neurons/layers increase power.** Each additional ReLU neuron adds another thresholded feature; combining many of them lets the network approximate complex relationships by stitching together many local, piecewise-linear behaviors.