INF721
2023/2

UFV

Aprendizado em Redes Neurais Profundas

A9: Regularização

Logística

Avisos

- Entrega do Projeto P2: Multilayer Perceptron adiada para o dia 25/09

Última aula

- MLP em numpy

Plano de Aula

- Experimentos com RNAs
- Normas Vetoriais e Matriciais
- Reguralização L1
- Regularização L2
- Dropout

Experimentos com RNAs

O processo de treinamento de uma RNA com gradiente descendente pode gerar diferentes resultados:

Erro de Treinamento	Alto	Baixo	Baixo
Erro de Validação	Alto	Alto	Baixo
	Subajuste (alto viés)	**Sobreajuste (alta variância)**	**Ajuste Adequado**
			Objetivo final!

UFV

Experimentos com RNAs

Classificação de imagens de gatos vs. cachorros

Assumindo o ser humano como baseline, erro de previsão ~0%

Erro de Treinamento	15%	1%	1%
Erro de Validação	16%	11%	1%
	Subajuste (alto viés)	**Sobreajuste (alta variância)**	**Ajuste Adequado**

Objetivo final!

UFV

Subajuste (alto viés)

Durante a fase de treinamento, esse é o **primeiro** problema a ser resolvido e as possíveis soluções são:

- Aumentar o tamanho (capacidade) da rede
  - Número de camadas
  - Número de neurônios por camada
- Treinar pois mais tempo (aumentar o número de épocas)
- Outras arquiteturas (e.g., convoluionais, recorrentes)

Sobreajuste (alta variância)

Durante a fase de treinamento, esse é o **segundo** problema a ser resolvido e as possíveis soluções são:

- Coletar mais dados
- Regularização
- Outras arquiteturas (e.g., convoluionais, recorrentes)

Regularização

Simplificar modelos de aprendizado com o objetivo de reduzir sobreajuste:

- Regularização L2
- Regularização L1
- Dropout
- Treinar por menos tempo (*early stopping*)
- Aumentar conjunto de dados

Normas Vetoriais

Em Algebra Linear, uma **norma** é um função $||\cdot|| : X \to R^+$ que associa um vetor a um número real não-negativo com as seguintes propriedades:

Para quaisquer vetores $x, y \in X$ e $\alpha \in R$:

1. $||\cdot|| \geq 0$ e $||x|| = 0$ se $x = 0$
2. $||x + y|| \leq ||x|| + ||y||$
3. $||\alpha x|| = |\alpha| ||x||$

UFV

Normas Vetoriais l_p

Normas l_p são um tipo especial de norma, definidas da seguinte forma:

$$\|x\|_p = (\sum_{i=1}^{n} |x_i|^p)^{\frac{1}{p}}$$

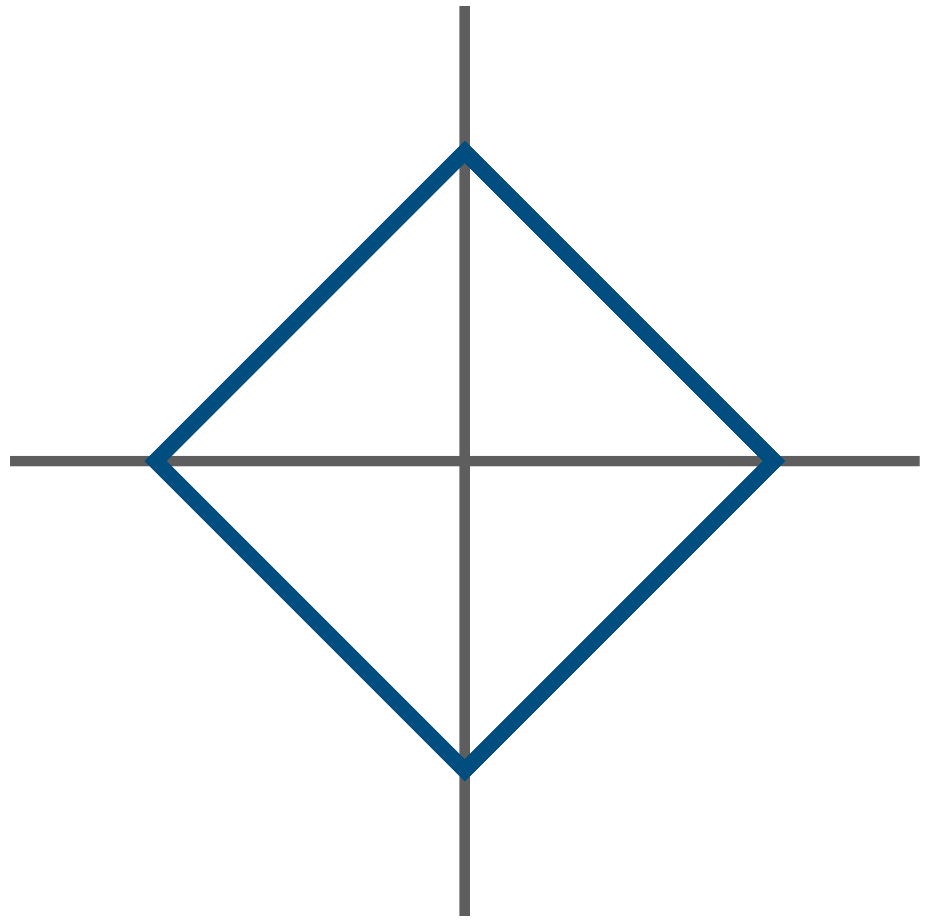
Duas normas l_p muito populares são:

Norma $\|x\|_1 = (\sum_{i=1}^{n} |x_i|^1)^{\frac{1}{1}} = (\sum_{i=1}^{n} |x_i|)$

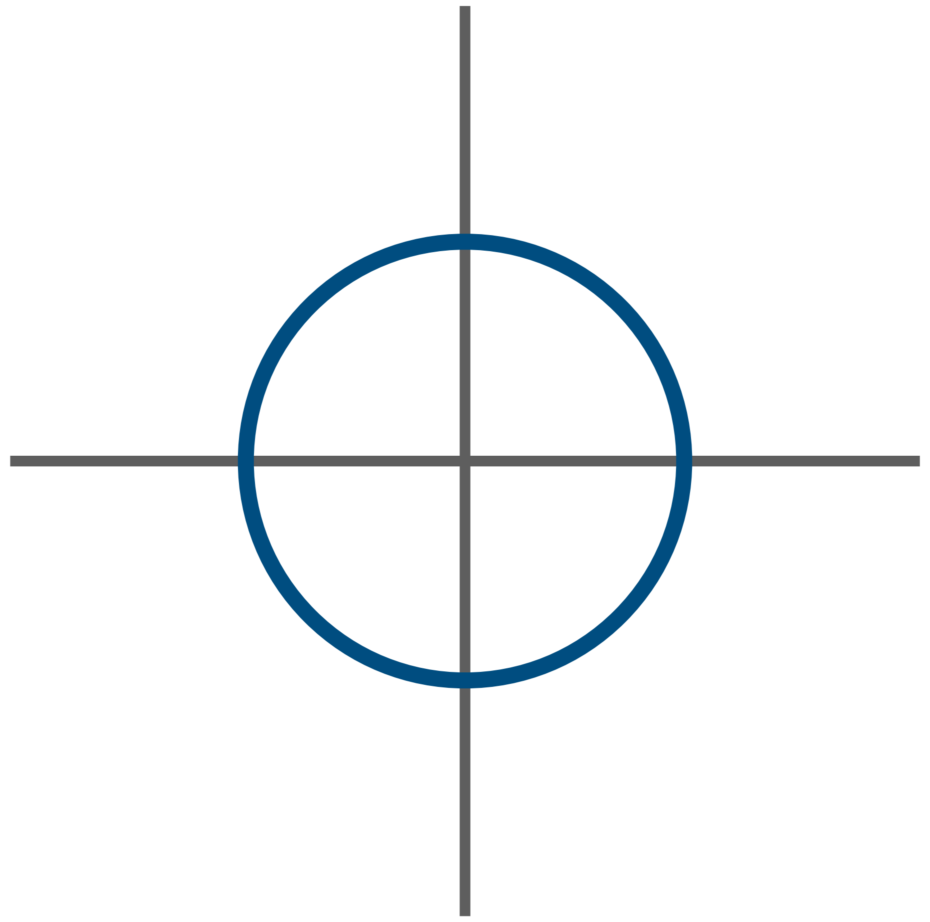
Norma (Euclidiana) $\|x\|_2 = (\sum_{i=1}^{n} |x_i|^2)^{\frac{1}{2}} = \sqrt{(\sum_{i=1}^{n} |x_i|^2)}$

Representação Geométrica das Normas Vetoriais l^p

Círculo unitário $(x \in R^2 : \|x\| = 1)$ para as normas vetoriais $\|x\|_1$ e $\|x\|_2$:



$$\|x\|_1 = (\sum_{i=1}^{n} |x_i|)$$



$$\|x\|_2 = \sqrt{(\sum_{i=1}^{n} |x_i|^2)}$$

Normas Matriciais

As normas matricais associam uma matriz a um número real não-negativo com as mesmas propriedades das normas vetoriais. As normas matriciais $\|\cdot\|_p$ tratam uma matriz $m \times n$ como um vetor com mn dimensões:

$$\|A\|_p = \left(\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^p\right)^{\frac{1}{p}}$$

Assim, duas $\|\cdot\|_p$ muito populares são:

Norma 1 $\|A\|_1 = \sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|$

Norma 2 (Frobenius) $\|A\|_2 = \sqrt{\left(\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2\right)}$

Regularização L2

A regularização L2 adiciona **o quadrado da norma** $||\cdot||_2$ na função de perda para penalizar RNAs com valores de pesos muito altos.

$$L(h) = -\frac{1}{n}\sum_{i=1}^{n} L(y^{(i)}, \hat{y}^{(i)}) + \frac{\lambda}{2n}||W||_2^2$$

Na regressão logística, usamos a norma vetorial ao invés da matricial!

onde λ é um hiperparâmetro que controla a intensidade da penalização.

Regularização L1

A regularização L2 adiciona **a norma $||\cdot||_1$** na função de perda para penalizar RNAs com valores de pesos muito altos.

$$L(h) = -\frac{1}{n}\sum_{i=1}^{n} L(y^{(i)}, \hat{y}^{(i)}) + \frac{\lambda}{2n}||W||_1$$

Na regressão logística, usamos a norma vetorial ao invés da matricial!

onde λ é um híperparâmetro que controla a intensidade da penalização.

A regularização L1 faz com que a matriz de pesos W seja esparsa!

Efeito da Regularização L2

Atualização de pesos com regularização:

$$W^{[l]} = W^{[l]} - \alpha\boxed{(dW^{[l]} + \frac{\lambda}{n}W^{[l]})}$$

Derivada parcial da função de erro regularizada com relação a W^l

$$= W^{[l]} - \frac{\alpha\lambda}{n}W^{[l]} - \alpha dW$$

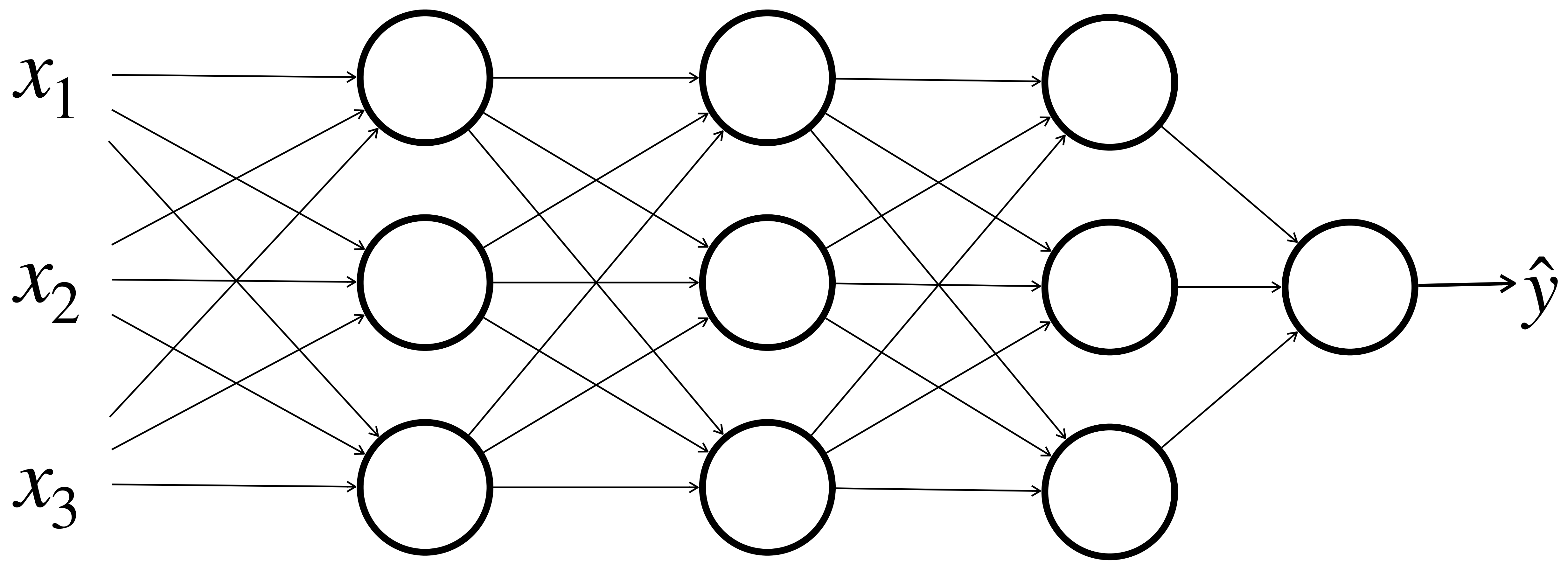
$$= \underbrace{(1 - \frac{\alpha\lambda}{n})}_{<1}W^{[l]} - \alpha dW$$

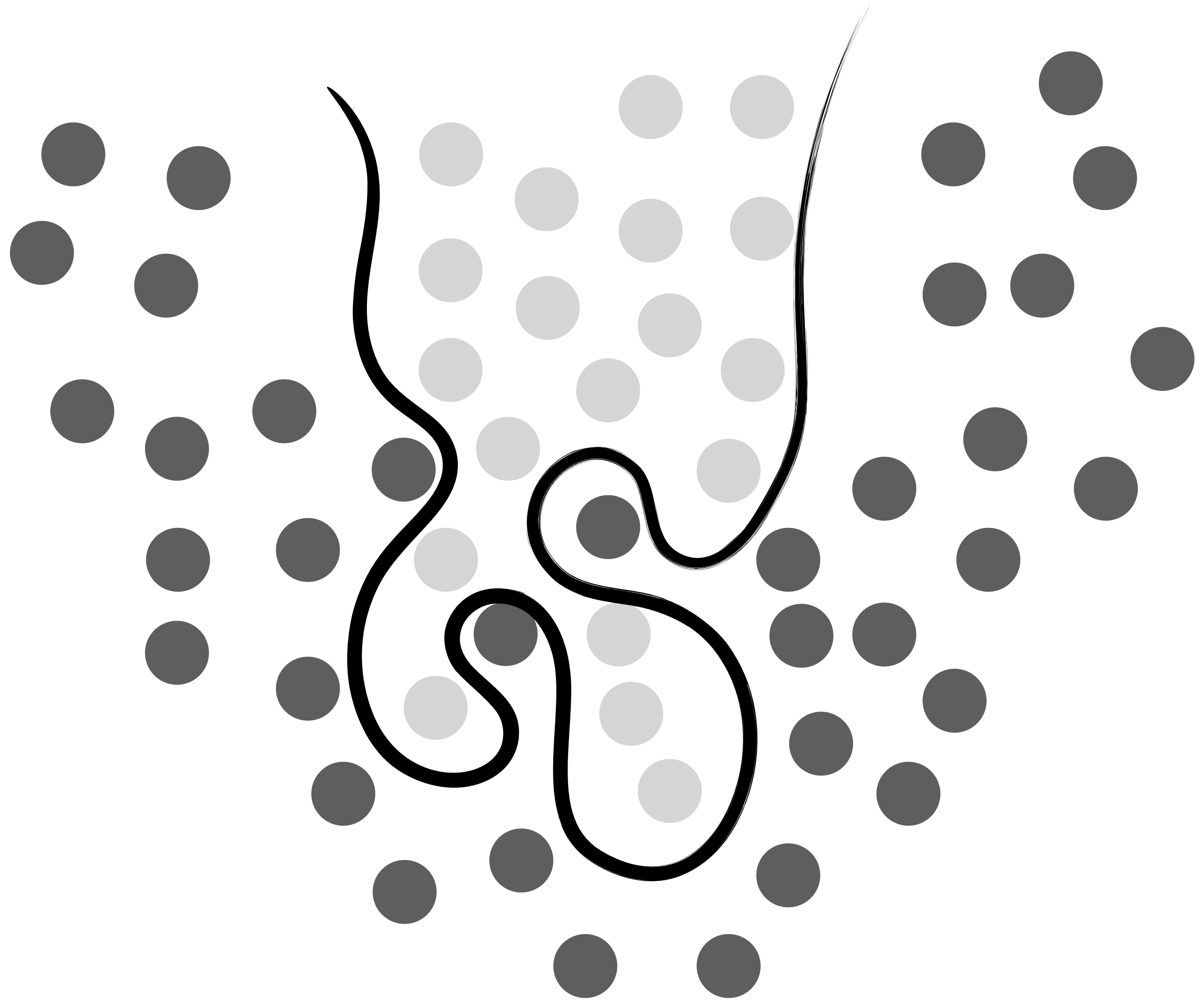
$< 1 \longrightarrow$ A regularizações L2 reduz os valores dos pesos $W^{[l]}$ e por isso, também é chamada de *Weight Decay*.

UFV

Efeitos da Regularização

$$L(h) = -\frac{1}{n}\sum_{i=1}^{n} L(y^{(i)}, \hat{y}^{(i)}) + \frac{\lambda}{2n}\|W\|_2^2$$

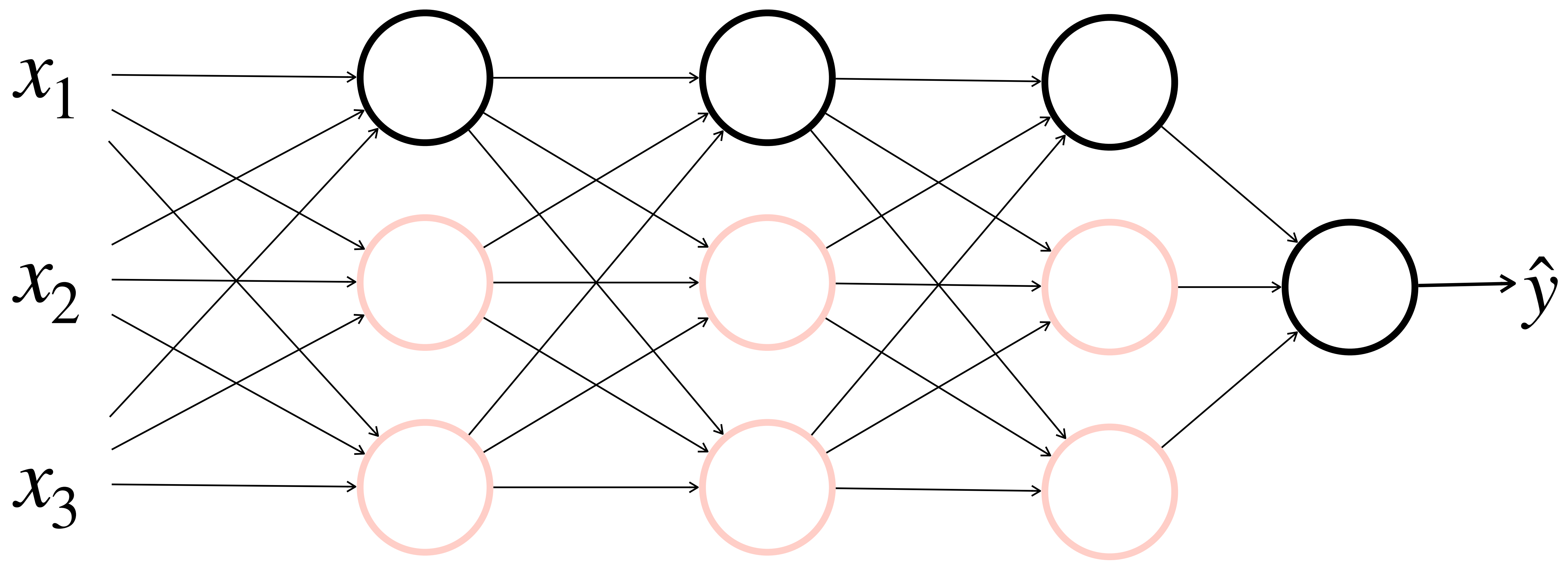


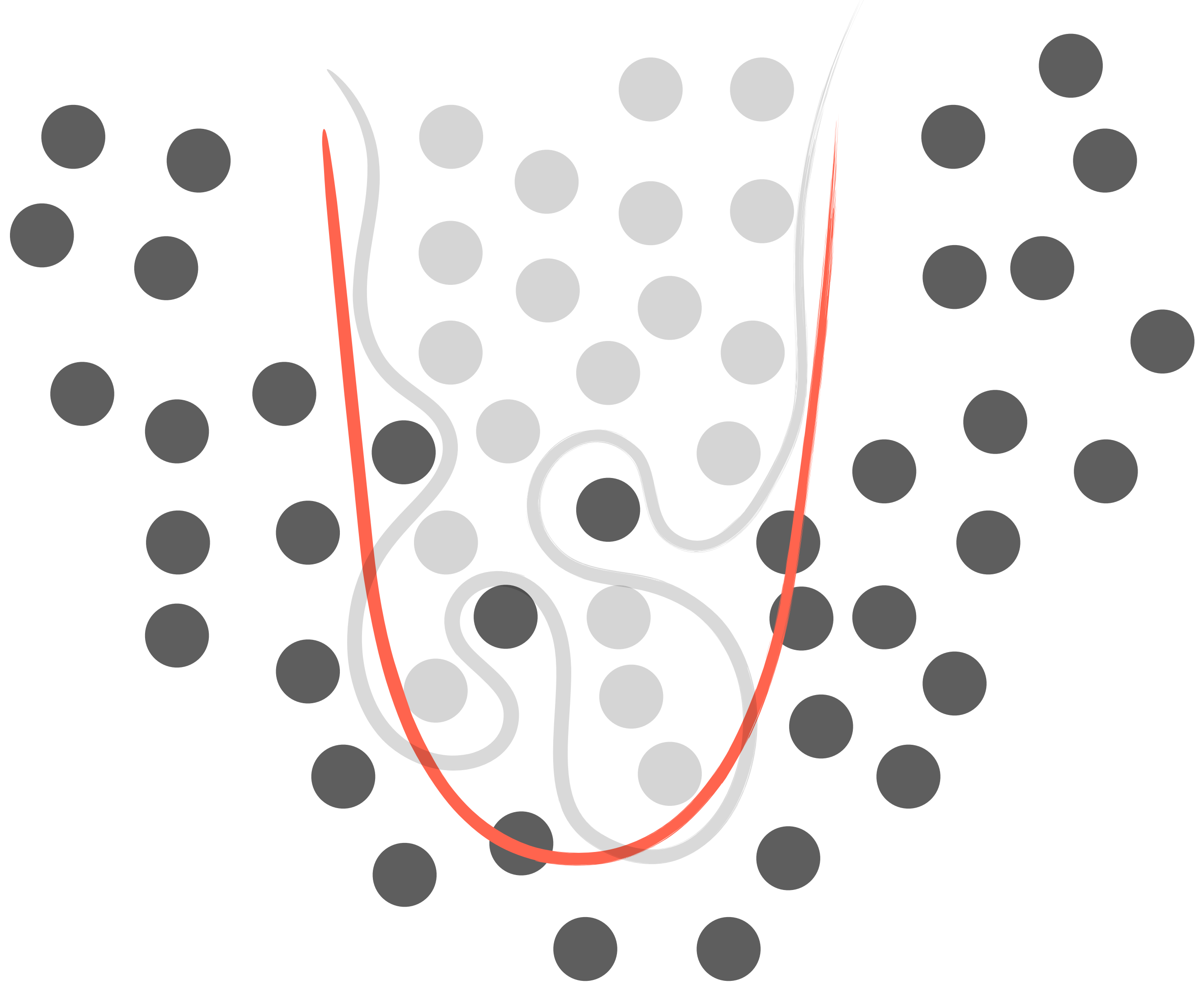


UFV

Efeitos da Regularização

$$L(h) = -\frac{1}{n}\sum_{i=1}^{n} L(y^{(i)}, \hat{y}^{(i)}) + \frac{\lambda}{2n}\|W\|_2^2 \longrightarrow W^{[l]} \approx 0$$

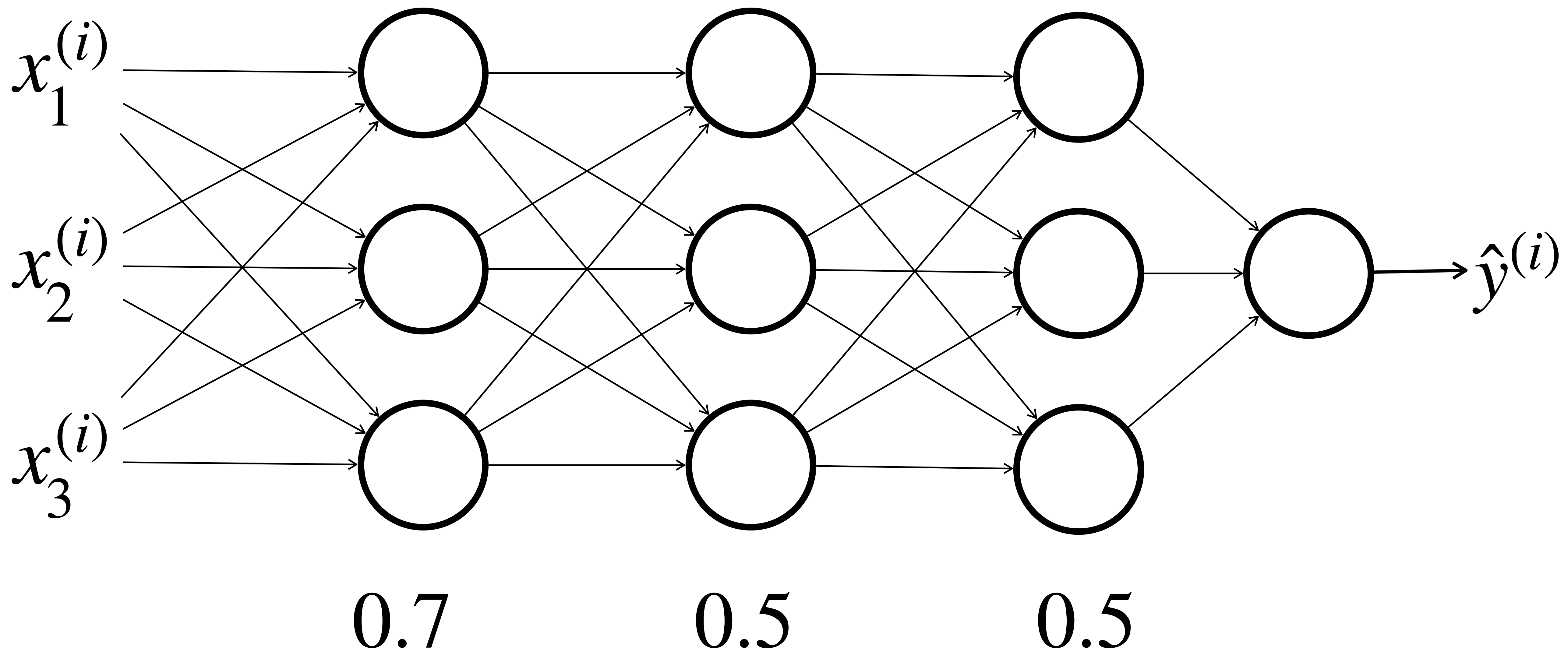




Ao reduzir os pesos de alguns neurônios, a regularização simplifica a hipótese de uma RNA em tempo de treinamento, tornando a fronteira de decisão mais simples também.

Dropout

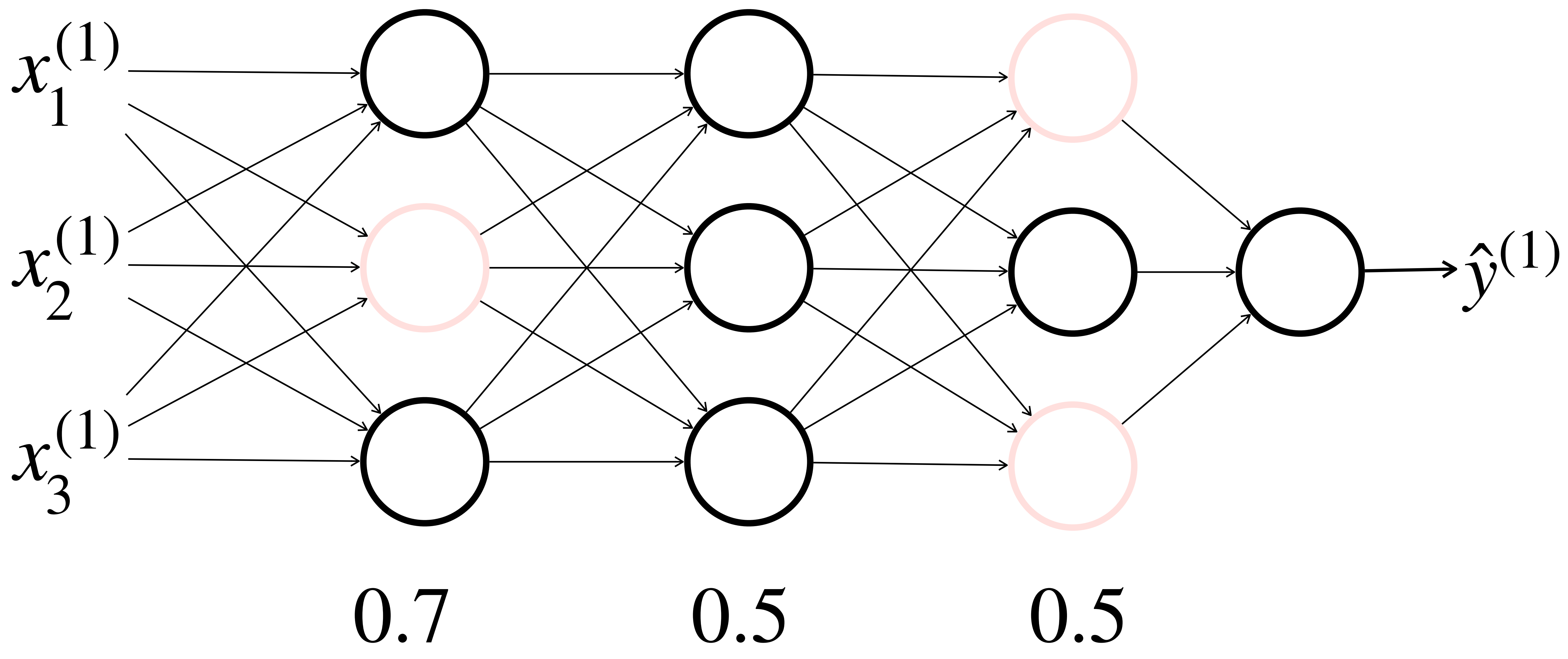
Dropout é uma técnica de regularização que desabilita neurônios aleatórios antes de calcular o erro para cada exemplo do conjunto de treinamento.



Cada camada recebe uma probabilidade de manter os neurônios naquela camada ativos antes do cálculo do erro para cada exemplo (i).

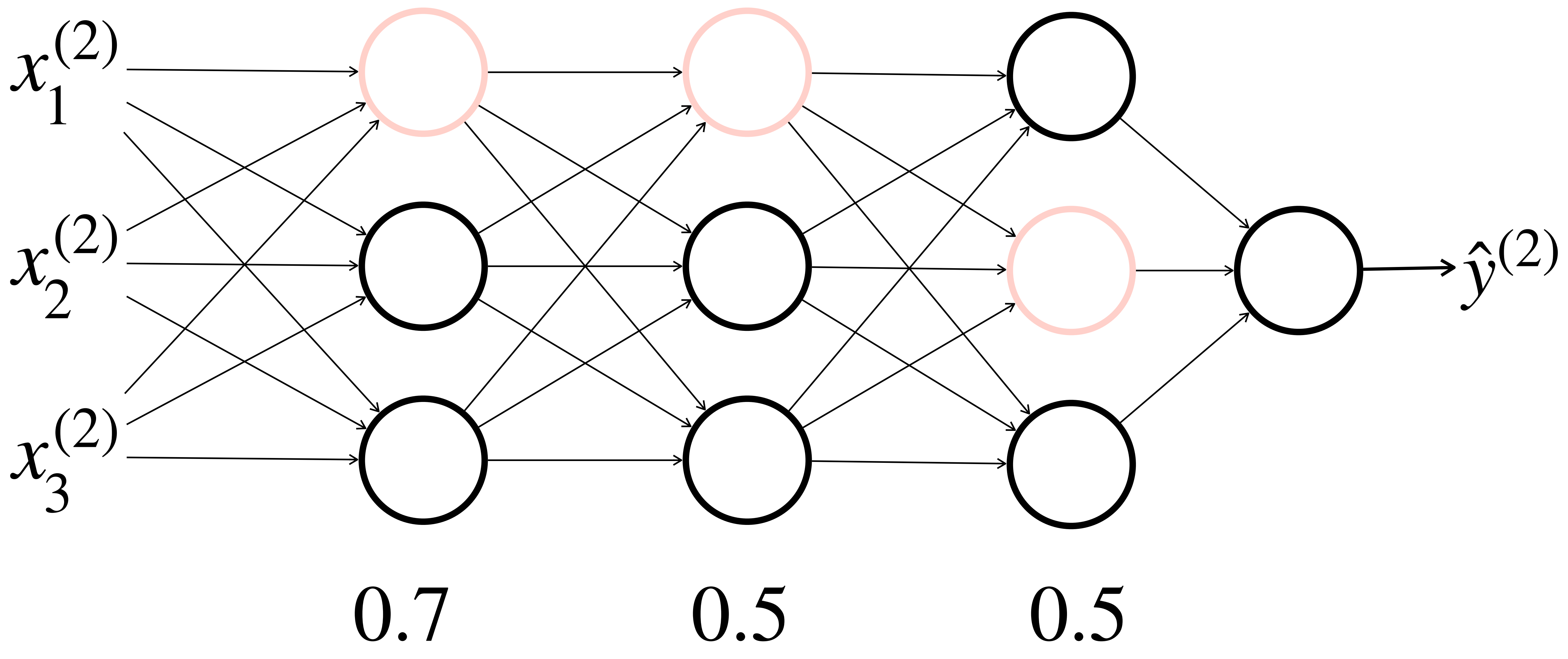
Dropout

Dropout é uma técnica de regularização que desabilita neurônios aleatórios antes de calcular o erro para cada exemplo do conjunto de treinamento.



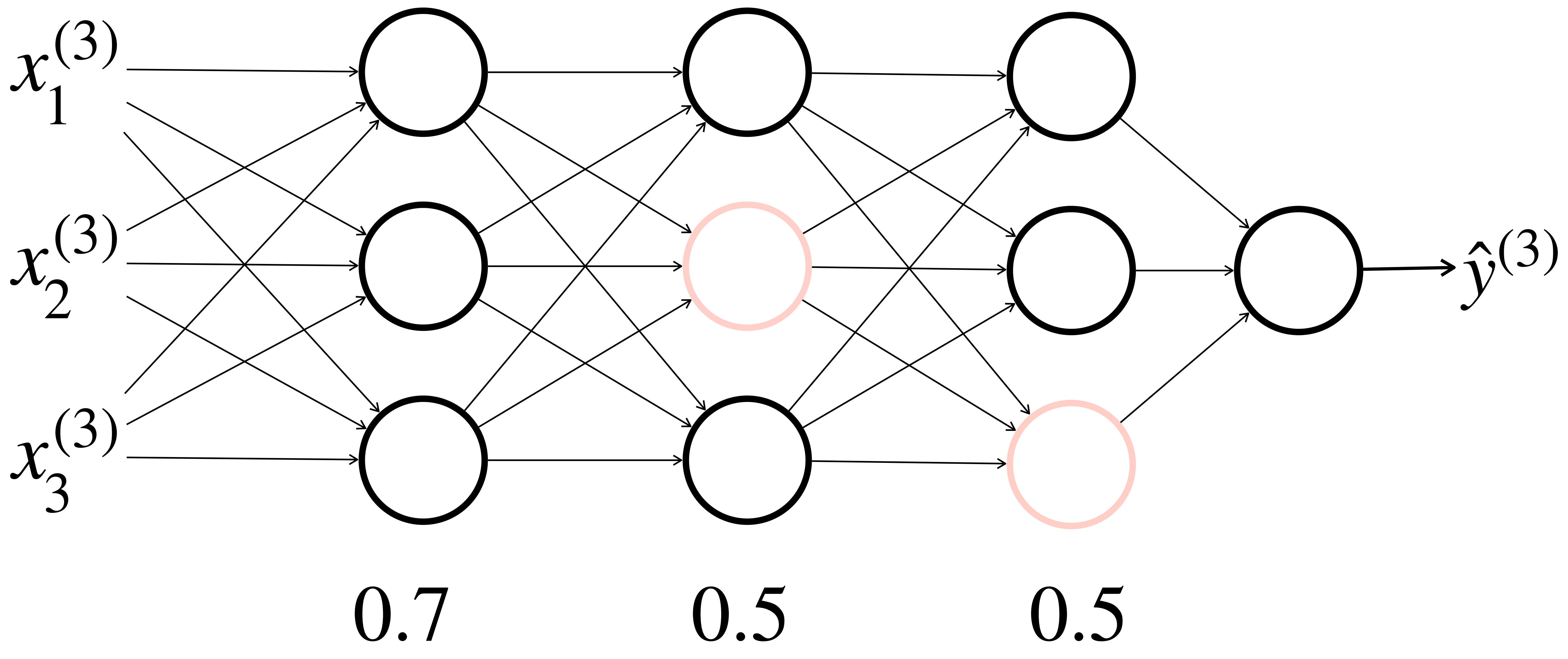
Dropout

Dropout é uma técnica de regularização que desabilita neurônios aleatórios antes de calcular o erro para cada exemplo do conjunto de treinamento.



Dropout

Dropout é uma técnica de regularização que desabilita neurônios aleatórios antes de calcular o erro para cada exemplo do conjunto de treinamento.



Uma configuração de RNA diferente é treinada para cada exemplo (i), forçando uma distribuição de pesos entre os neurônios de uma camada de maneira mais uniforme, não em apenas uma ou poucas entradas.

Próxima aula

A10: Otimização

Algoritmos de otimização avançados: Mini-batch, Momentum, RMSProp, e Adam.