

Redes neuronales y el mundial



IA

- robots vs programas ◀
- tipos, límites borrosos ◀
- para qué? ◀
- humor vs realidad ◀

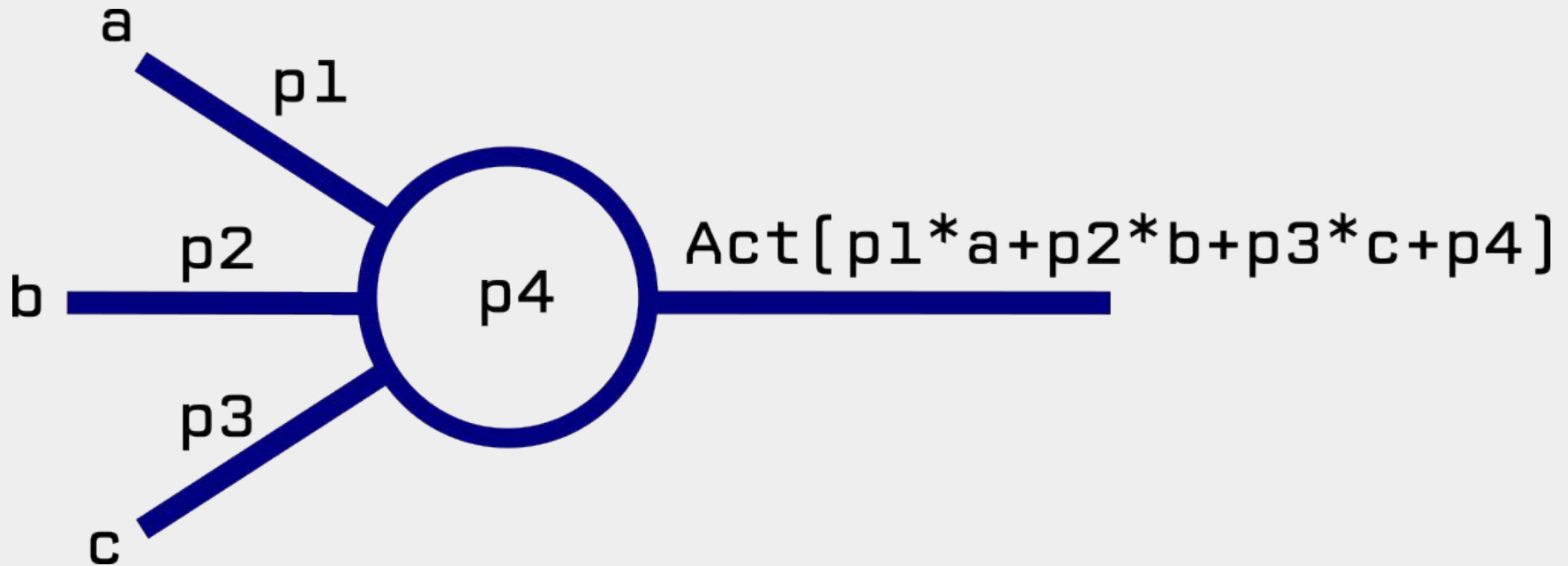
Redes Neuronales

Redes Neuronales

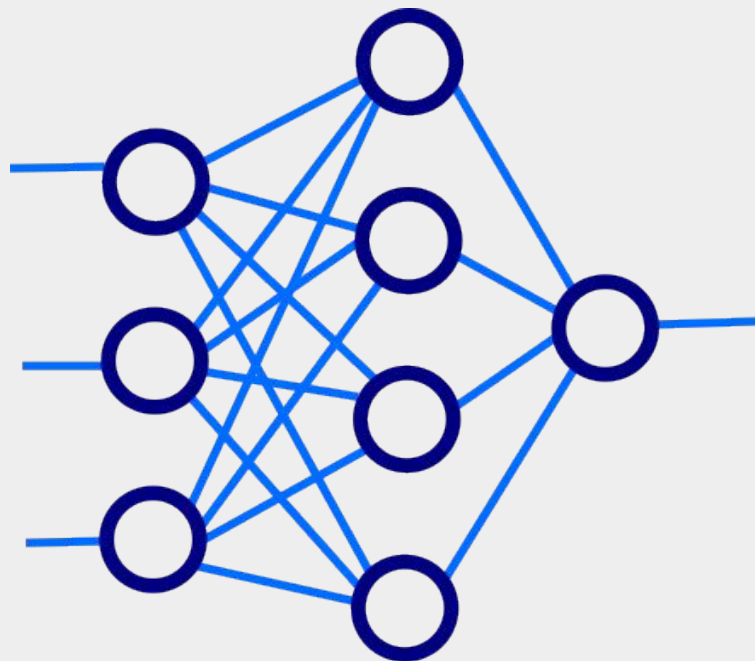
una técnica para crear IAs que aprenden ◀
aprenden a partir de datos, generalizan ◀
muy muy poderosas ◀

Cómo aprenden?

Cómo aprenden?



Cómo aprenden?



Y el mundial??

Y el mundial??

gente que patea una pelota ◀
pelota en el arco del otro equipo → bueno ◀
qué país tiene los mejores peloteadores? ◀

Y el mundial??

problema de “juguete”, pero caso real ◀
discusión familiar: predecir fútbol con ML ◀

Y el mundial??

- problema de “juguete”, pero caso real ◀
- discusión familiar: predecir fútbol con ML ◀
- prode familiar! **el último cocina** ◀

Mundial + Redes neuronales

Mundial + Redes neuronales

entradas



salida ◀

Mundial + Redes neuronales

entradas
quiénes juegan



salida ◀
quién gana

Mundial + Redes neuronales

entradas
quiénes juegan



salida ◀
quién gana

aprender a partir de ejemplos pasados ◀

Paso 1: datos

Paso 1: datos

suele renegarse ◀

Paso 1: datos

suele renegarse ◀

Wikipedia!... html no estandarizado ◀

Paso 1: datos

suele renegarse ◀
Wikipedia!... html no estandarizado ◀
Python! scrapeo → csv ◀
y arreglos a mano (ej: países cambian...) ◀

Paso 1: datos

14 June 1998

Argentina 

1-0

 Japan

Jamaica 

1-3

 Croatia

20 June 1998

Japan 

0-1

 Croatia

21 June 1998

Argentina 

5-0

 Jamaica

26 June 1998

Argentina 

1-0

 Croatia

Japan 

1-2

 Jamaica

id	year	team1	score1	score2	team2
0	1950	Brazil	4	0	Mexico
1	1950	Yugoslavia	3	0	Switzerland
2	1950	Brazil	2	2	Switzerland
3	1950	Yugoslavia	4	1	Mexico
4	1950	Brazil	2	0	Yugoslavia
5	1950	Switzerland	2	1	Mexico
6	1950	England	2	0	Chile
7	1950	Spain	3	1	United States
8	1950	Spain	2	0	Chile
9	1950	United States	1	0	England

Paso 1: datos

texto vs función matemática ◀

Paso 1: datos

texto vs función matemática ◀
salida: winner -1 / 0 / 1 ◀
entradas: stats, orden ◀

Paso 1: datos

- texto vs función matemática ◀
- salida: winner -1 / 0 / 1 ◀
- entradas: stats, orden ◀
- sets de train vs test ◀

Paso 1: datos

texto vs función matemática ◀

salida: winner -1 / 0 / 1 ◀

entradas: stats, orden ◀

sets de train vs test ◀

Pandas y Scikit-Learn al rescate! ◀

Paso 1: datos

```
matches = matches[matches.score1 != matches.score2] # remove ties  
matches['winner'] = matches.score1 > matches.score2
```

```
train, test = train_test_split(matches, test_size=0.2)
```

Paso 2: primeros modelos

Paso 2: primeros modelos

Regresión Logística: **mal** ◀

Paso 2: primeros modelos

Regresión Logística: mal ◀

KNN: mal ◀

Paso 2: primeros modelos

Regresión Logística: mal ◀

KNN: mal ◀

Red Neuronal: overfit ◀

Paso 3: datos de vuelta

Paso 3: datos de vuelta

el problema de los empates ◀
salida binaria: winner 1 / 0

Paso 3: datos de vuelta

el problema de los empates ◀

salida binaria: winner 1 / 0

menos stats (prevenir overfit) ◀

Paso 3: datos de vuelta

	year	matches_won_percent	podium_score_yearly	matches_won_percent_2	podium_score_yearly_2	winner
1521	2014	0.391304	0.500000	0.735294	3.250000	False
1522	2014	0.428571	0.000000	0.307692	0.000000	True
1523	2014	0.375000	0.000000	0.250000	0.000000	False
1524	2014	0.470588	2.000000	0.685185	3.375000	True
1526	2014	0.409091	0.000000	0.000000	0.000000	True
1528	2014	0.666667	2.857143	0.357143	0.000000	True
1529	2014	0.111111	0.000000	0.333333	0.000000	False
1530	2014	0.762500	5.875000	0.272727	0.000000	True
1531	2014	0.444444	0.000000	0.590909	1.000000	False
1532	2014	0.500000	0.250000	0.222222	0.222222	True

Paso 4: red neuronal

Paso 4: red neuronal

PyBrain ◀

Paso 4: red neuronal

Paso 4: red neuronal

Keras/TensorFlow! ◀

Paso 4: red neuronal

```
network = Sequential([
    Input((5, )),
    Normalization(),
    Dense(10, activation='sigmoid'),
    Dense(10, activation='sigmoid'),
    Dense(1, activation='sigmoid'),
])

network.compile(
    optimizer='adam',
    loss='binary_crossentropy',
    metrics=['accuracy'],
)
```

Paso 4: red neuronal

entrenarla! ◀

Paso 4: red neuronal

```
network.fit(  
    train[input_cols], train[output_col],  
    epochs=50,  
    batch_size=128,  
)
```

Paso 5: evaluar

Paso 5: evaluar

accuracy: % de aciertos ◀

Paso 5: evaluar

accuracy: % de aciertos ◀
con los datos de train ◀

Paso 5: evaluar

accuracy: % de aciertos ◀
con los datos de train ◀
con los datos de test ◀

Paso 5: evaluar

accuracy: % de aciertos ◀
con los datos de train ◀
con los datos de test ◀
Scikit-learn :) ◀

Paso 5: evaluar

```
train_predictions = network.predict(train[input_cols])  
accuracy_score(train[output_col], train_predictions.round())  
74 %
```

```
test_predictions = network.predict(test[input_cols])  
accuracy_score(test[output_col], test_predictions.round())  
76 %
```

Paso 5: evaluar

Paso 5: evaluar

el piso es 50% eh ◀

75% es not great, not bad ◀

Paso 5: evaluar

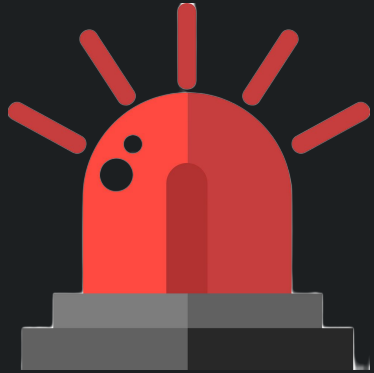
el piso es 50% eh ◀

75% es not great, not bad ◀

suficiente para no salir último ;) ◀

Paso 6: a producción

Paso 6: a producción



EL PRODE PIDE GOLES!!!

Paso 6: a producción

datos al rescate ◀

resultado más común: 2-1 ◀

Resultados en producción

Resultados en producción

prode familiar, 10 personas ◀

Resultados en producción

prode familiar, 10 personas ◀
1er puesto :)

Resultados en producción

prode familiar, 10 personas ◀
1er puesto :)

web El Ega, 250 personas ◀

Resultados en producción

prode familiar, 10 personas ◀
1er puesto :)

web El Ega, 250 personas ◀
1er puesto :D :D

Algunas moralejas

Algunas moralejas

probar temprano, iterar ◀

Algunas moralejas

probar temprano, iterar ◀
"good enough for prod" ◀

Algunas moralejas

probar temprano, iterar ◀

"good enough for prod" ◀

pensar cómo se va a usar en prod! ◀

Algunas moralejas

probar temprano, iterar ◀

"good enough for prod" ◀

pensar cómo se va a usar en prod! ◀

mirar la data puede ayudar mucho ◀

Algunas moralejas

probar temprano, iterar ◀

"good enough for prod" ◀

pensar cómo se va a usar en prod! ◀

mirar la data puede ayudar mucho ◀

no esperar a ser Google o OpenAI ◀

Muchas gracias!

bit.ly/neuronas_mundial

Fisa (Juan Pedro Fisanotti) | fisadev@gmail.com | [@fisadev](https://twitter.com/fisadev)