

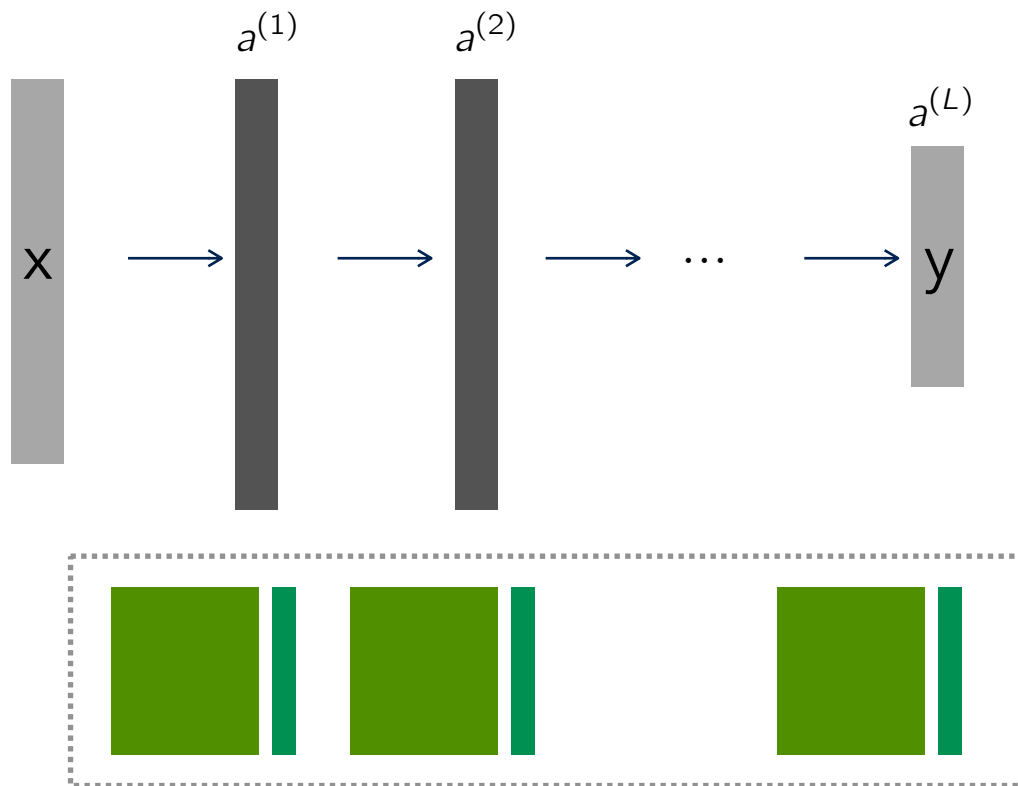
# Arquitectura Básica de Redes Neuronales

Clasificación y Regresión



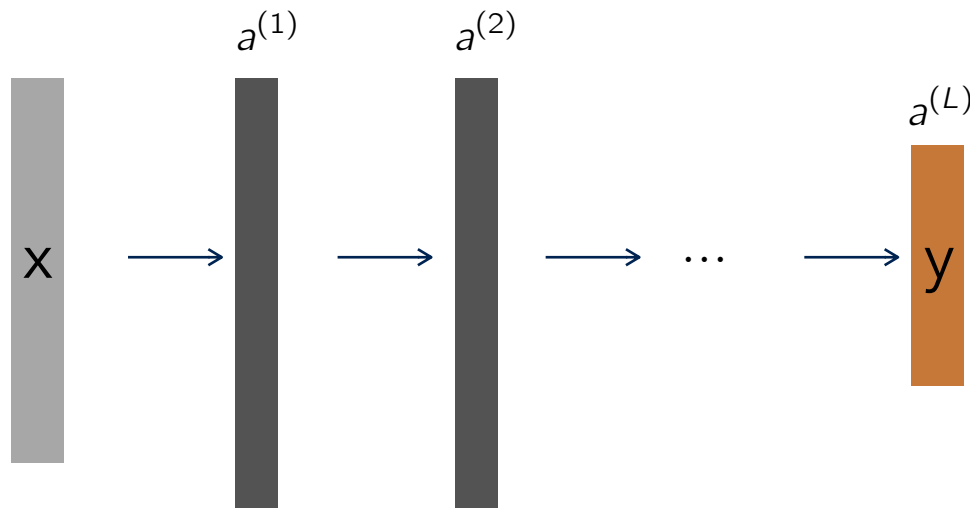
# Redes FF Estándar

- Una red feed-forward implementa una función entre el espacio de entrada  $X$  y el espacio de salida  $Y$  componiendo una serie de transformaciones vector-vector **con parámetros entrenables**.



# Especialización de la Capa de Salida

- Como la capa de salida produce el resultado que esperamos de la red, es necesario especializar esta capa a la tarea para la cual la red se está construyendo. Las tareas más comunes son: **regresión y clasificación.**

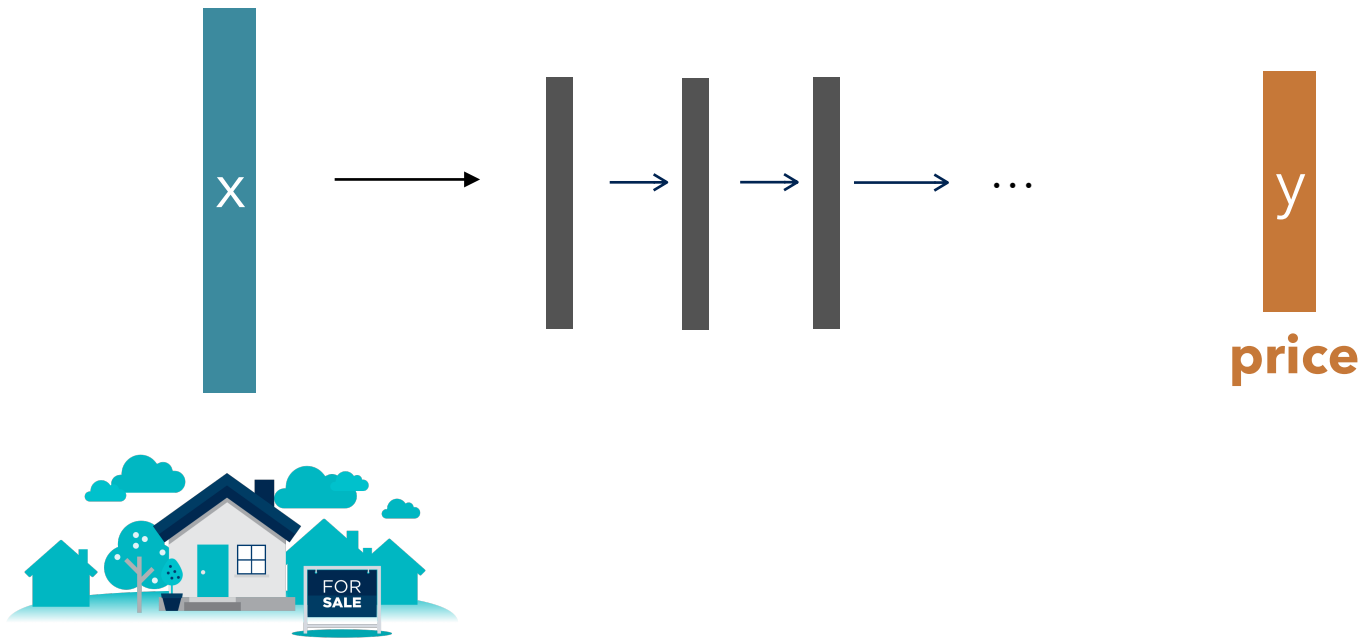


$x$

# Regresión

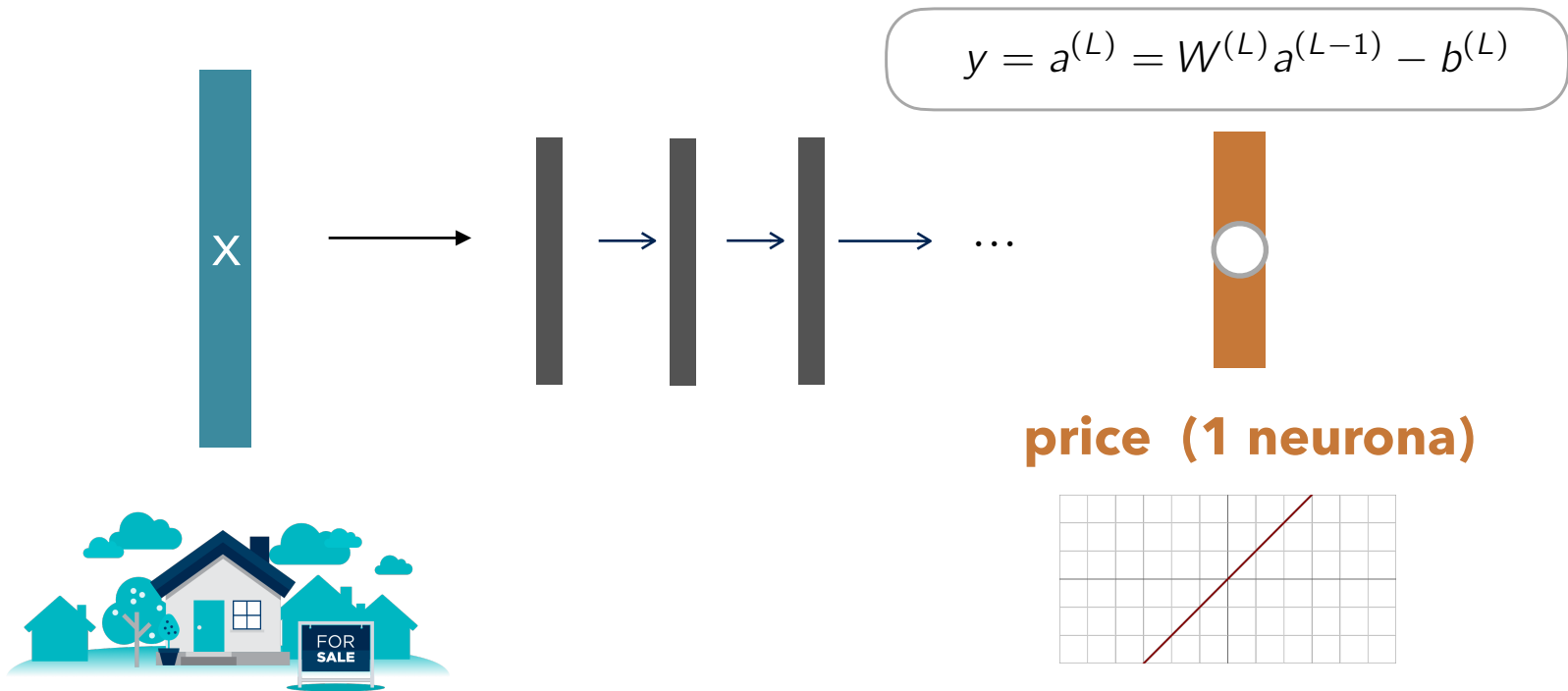
- En regresión, la salida es un número o un conjunto de números reales. Por ejemplo: el precio de un producto o la aceleración que debe darse a una rueda.

house descriptor



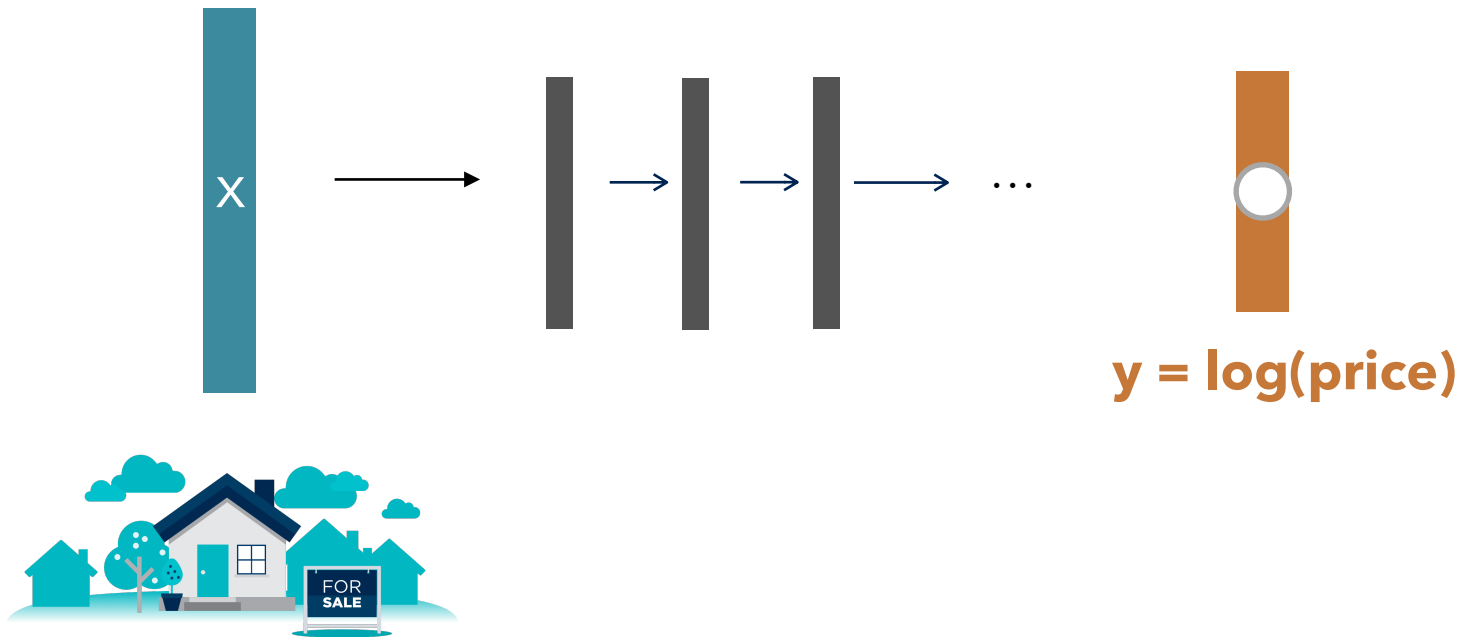
# Regresión

- En este caso, lo usual es utilizar un número de neuronas de salida igual a número de valores que queremos aproximar y emplear una función de activación lineal para la capa salida, de manera de poder cubrir el rango de posibles salidas:



# Regresión

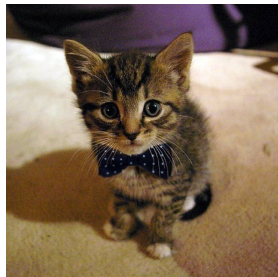
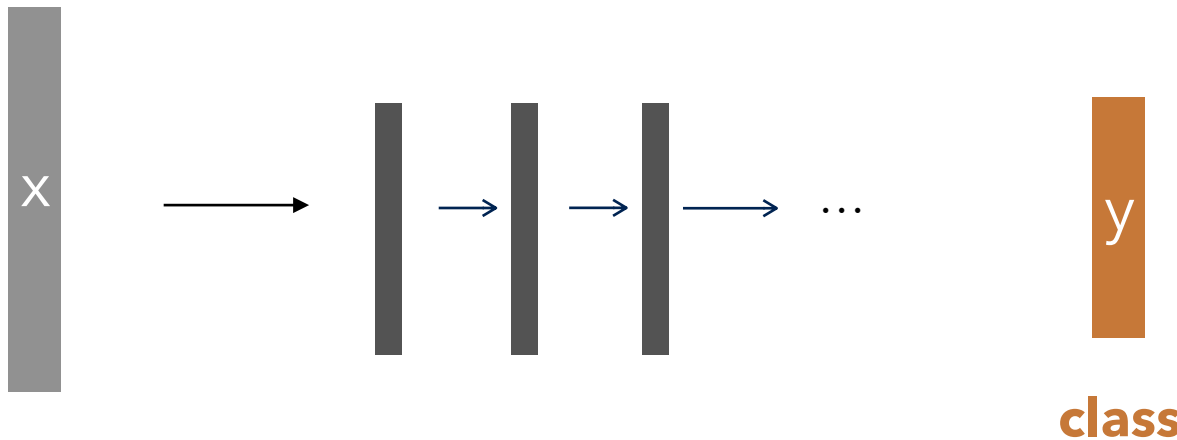
- No existe una metodología afirmada para manejar el caso en que la salida tiene un rango limitado en los reales. En el caso de valores positivos, es común considerar una transformación logística para recuperar el rango de todos los reales.



# Clasificación

- En clasificación, la salida es una categoría de un conjunto finito y sin estructura más rica que la de un conjunto finito. Por ejemplo: queremos distinguir entre imágenes de humanos (H), gatos (G) o perros (P).

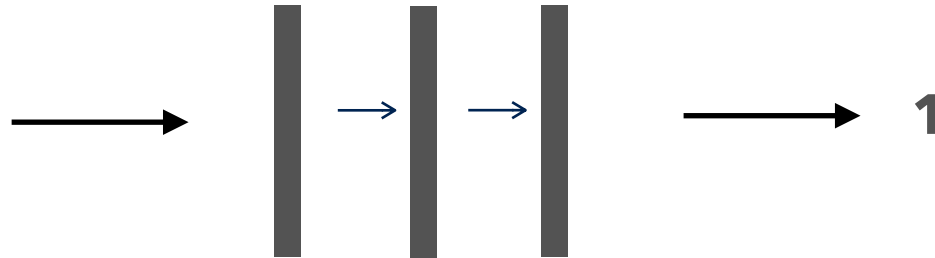
image descriptor



# Clasificación

- Un enfoque ingenuo consiste en representar las categorías como números y utilizar una red para predecir ese número.
- En general esto **no** es una buena idea. Los códigos son arbitrarios y no son operables como números reales.

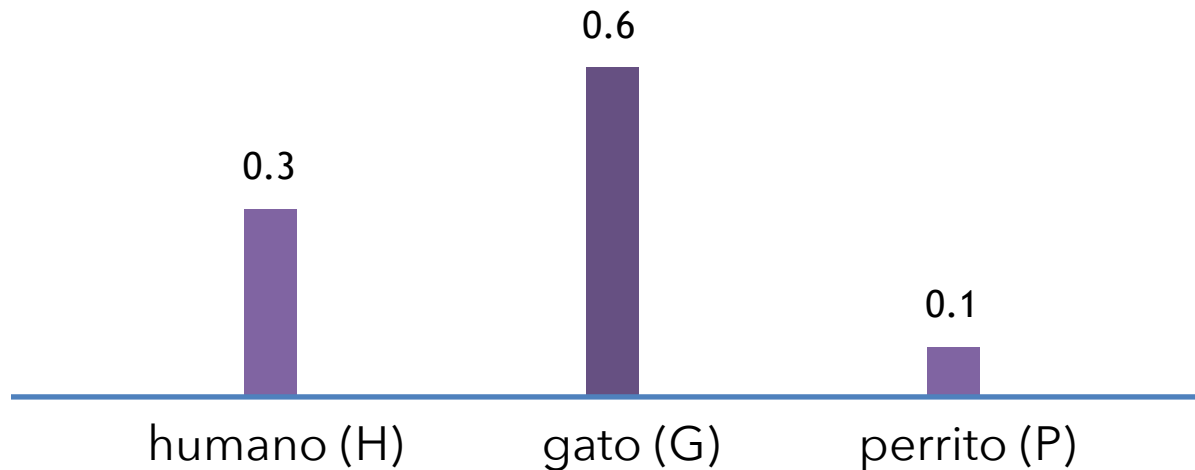
**Cat: 1, Dog: 2, Human: 3**





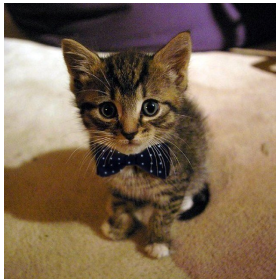
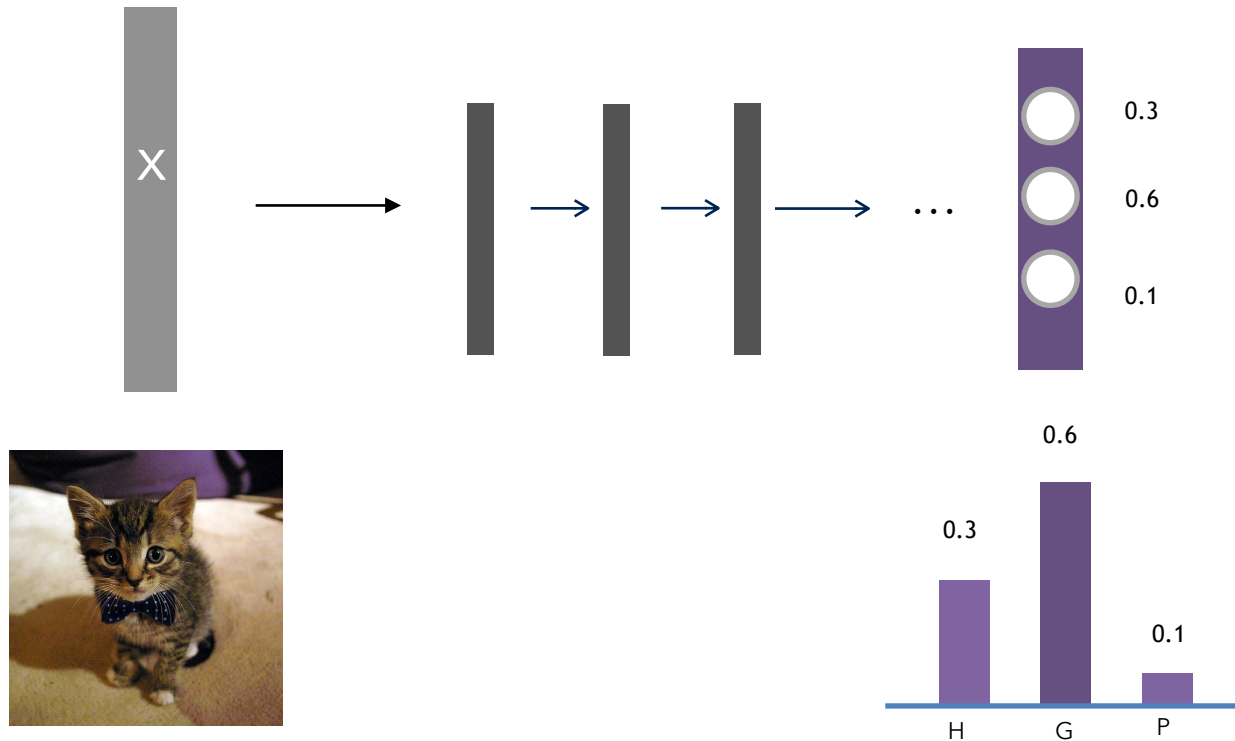
# Clasificación

- Una metodología muy efectiva en la práctica es utilizar la red para predecir una **distribución de probabilidad** sobre las posibles clases.



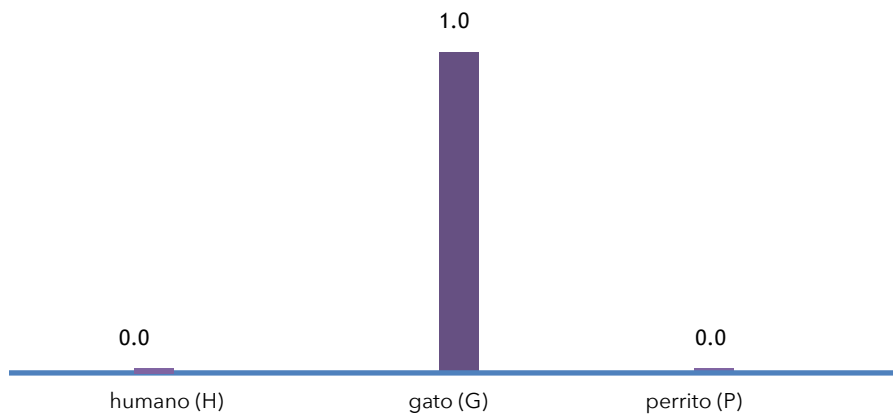
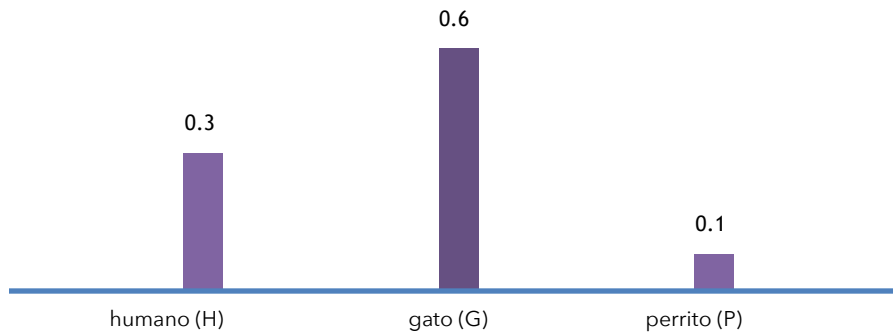
# Clasificación

- Para esto, basta utilizar un **número de neuronas igual al número de clases** y elegir una función de activación que garantice que las salidas son interpretables como una distribución de probabilidad (es decir, los valores están en  $[0,1]$  y suman 1).



# Clasificación

- Una metodología muy efectiva en la práctica es utilizar la red para predecir una **distribución de probabilidad** sobre las posibles clases.

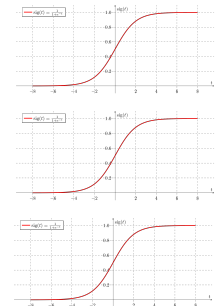
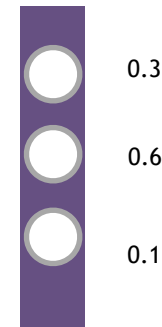
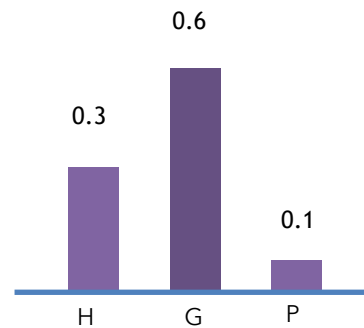


# Capa Softmax

- Un modo muy común de lograr esto es utilizar en la salida una función de activación especial denominada **softmax**:

$$y = a^{(L)} = g\left(W^{(L)}a^{(L-1)} - b^{(L)}\right)$$

$$g(z) = \frac{\exp(z)}{\sum_{k=1}^K \exp(z_k)}$$



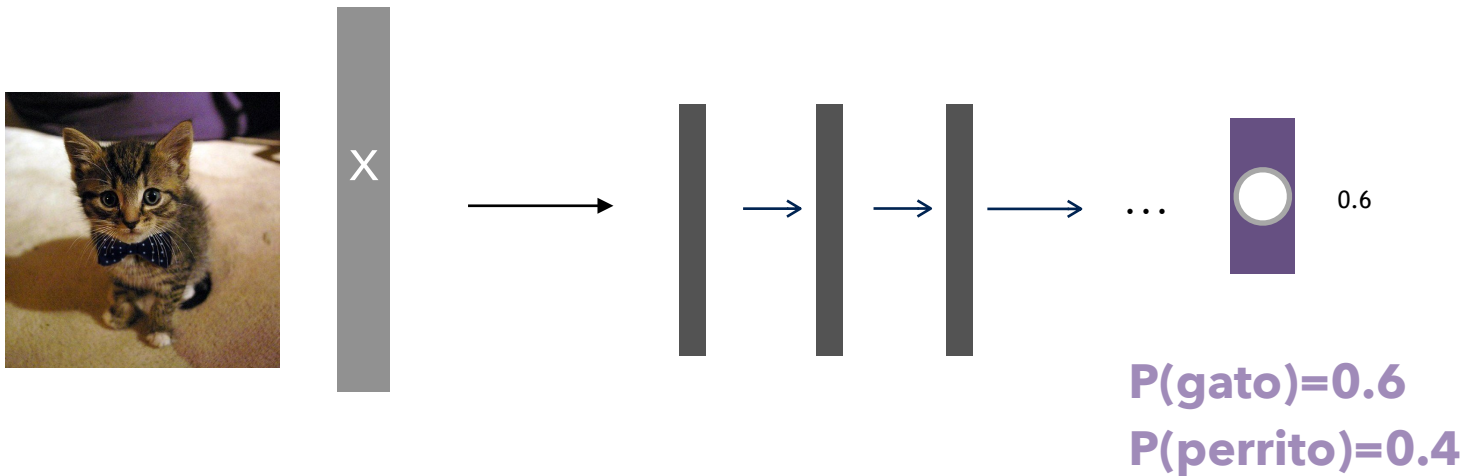
- Notemos que se trata esencialmente de una función de activación logística normalizada.

$$p(y = 1) = g(z) = \frac{1}{1 + \exp(-z)} = \frac{\exp(z)}{1 + \exp(z)}$$
$$p(y = 0) = 1 - g(z) = \frac{1}{1 + \exp(z)}$$

# Casos Especiales

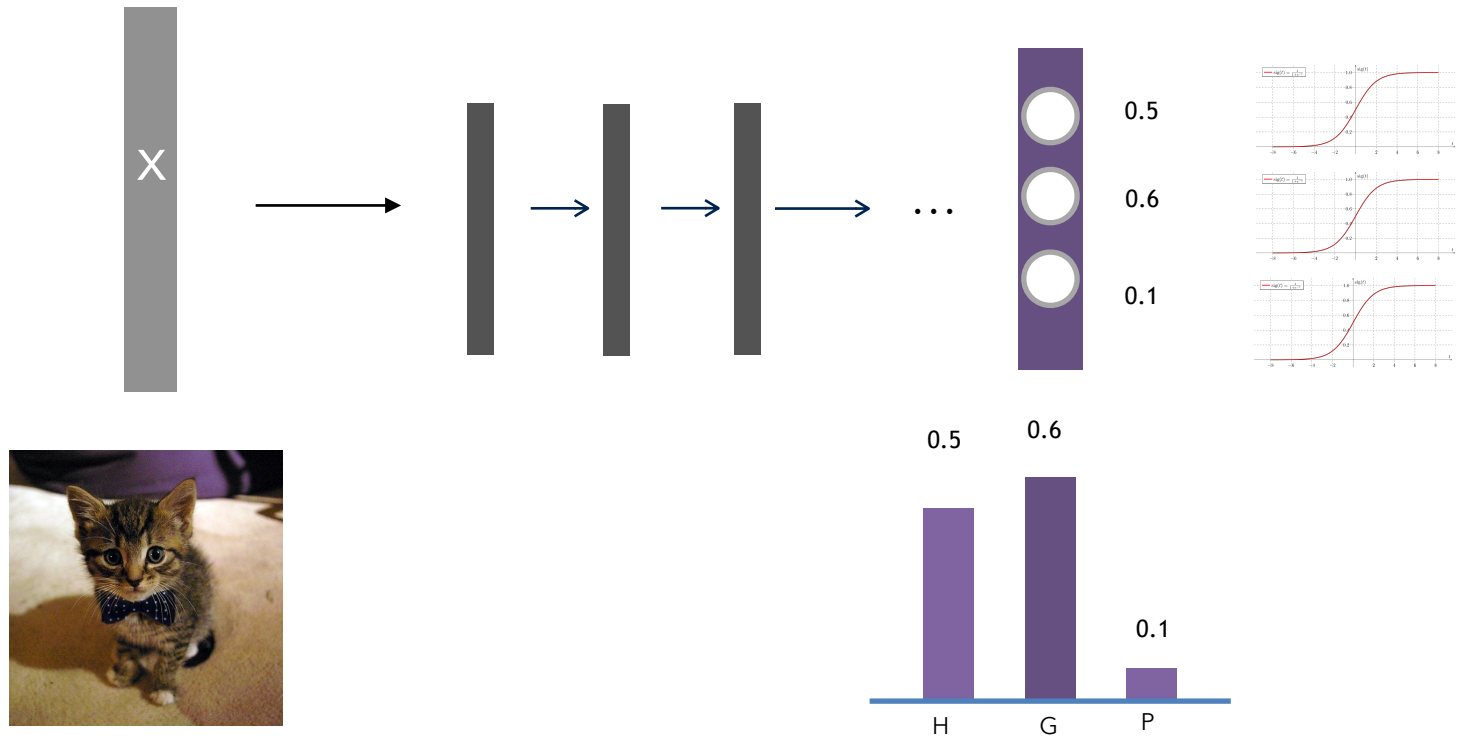
- Del comentario anterior, es claro que en el caso en que se tienen  $K=2$  clases, usar 2 neuronas de salida es redundante: 1 salida permite predecir exactamente la otra.
- En este caso bastaría usar una única neurona sigmoideal en la salida.

**clases posibles = {gato, perrito}**



# Casos Especiales

- Otro caso especial es la situación en que se tienen clases no excluyentes (multi-label classification). En este caso, las probabilidades no necesariamente deben sumar 1 y por lo tanto basta usar funciones de activación sigmoidales para cada neurona.



## Entonces ...

- La capa de salida de una red debe producir el valor deseado usando una codificación apropiada para el problema. Ello implica **especializar la capa de salida** con una forma y/o activación característica.
- **En regresión**, se suele usar una capa salida con tantas neuronas como valores que queremos aproximar y **función de activación lineal**.
- **En clasificación** con más de 2 categorías excluyentes, se suele usar una capa salida con tantas neuronas como clases y **función de activación softmax**.
- Tarea: Escriba un programa que visualice las probabilidades que salen de una capa softmax.