

ATACANDO REDES NEURONALES

SANTIAGO ANDRADE

WHOAMI

Desarrollador web, especializandome
en IA y ciberseguridad.



/andradefs



/andrade-fs



VIABILIDAD DE IMPACTO

Las redes neuronales están en un sinfín de tareas que realizamos a diario. Algunos casos sencillos de "adversarial attack" o "Backdoors"

Detección de spam

El más sencillo podemos encontrarlo en los comienzos de la detección de correos basura, clasificadores estándares como Naive Bayes tuvieron mucho éxito contra emails que contenían textos como: ¡Haga dinero rápido!, Refinancia tu hipoteca...

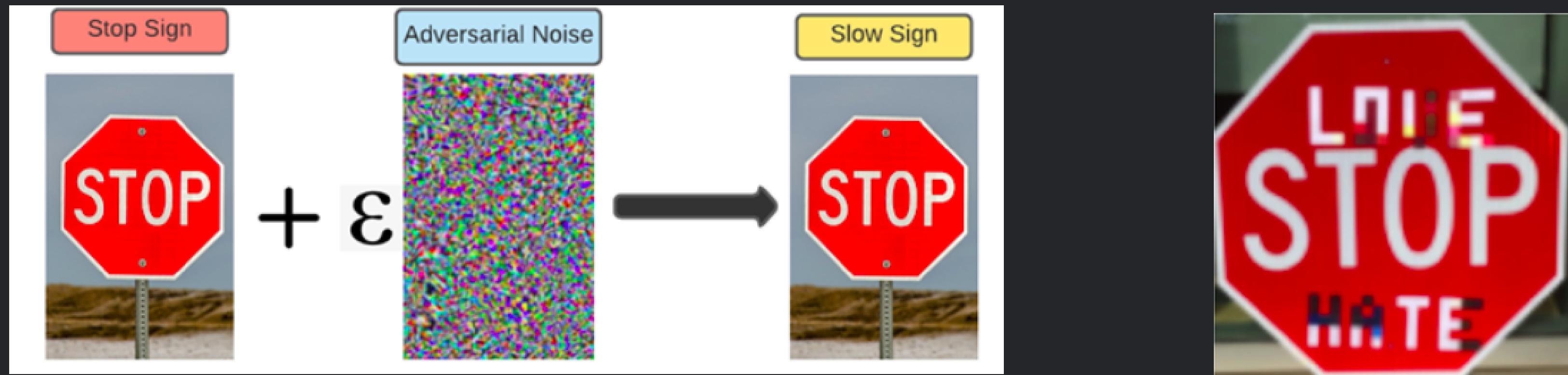
Así comenzaron a usar “disfraces” como: ¡H4G4 D|nero r4p1d0!...



O simplemente embebiendo el mensaje en una imagen...

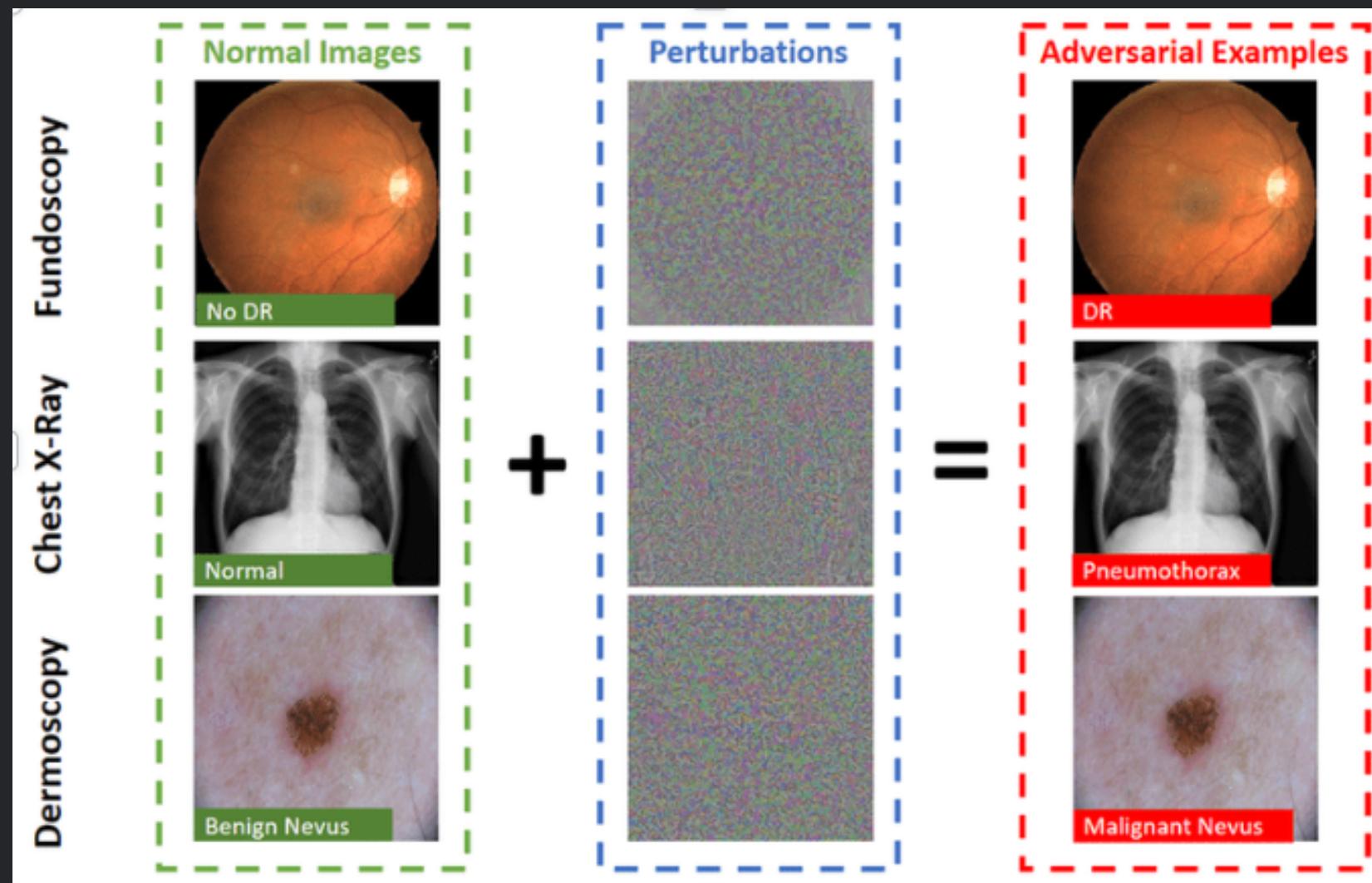
PILOTO AUTOMÁTICO

la categorización de los elementos que rodean al automóvil es uno de los componentes del sistema de aprendizaje profundo, que le permite desplazarse con seguridad y obedecer las leyes de la carretera distinguiendo personas, bicicletas, señales de tráfico y otros objetos.



MEDICINA

Un ataque de adversario no significa siempre algo malo, en este caso sería posible detectar lunares malignos, canceres, etc.. Cuando a simple vista es imposible detectar el cambio



OBJETIVO

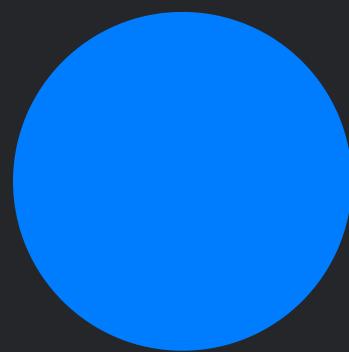
Adversarial Attack, Backdoors

Programación regular VS automática

Transformamos entradas en resultados mediante algoritmos

Programación regular

Entradas



Reglas y lógica



Resultado

5



Programación automática

Entradas



Reglas y lógica



Resultado



Escenario

Celsius



Fahrenheit

$$\text{Fahrenheit} = \text{Celsius} * 1.8 + 32$$

Escenario

Celsius



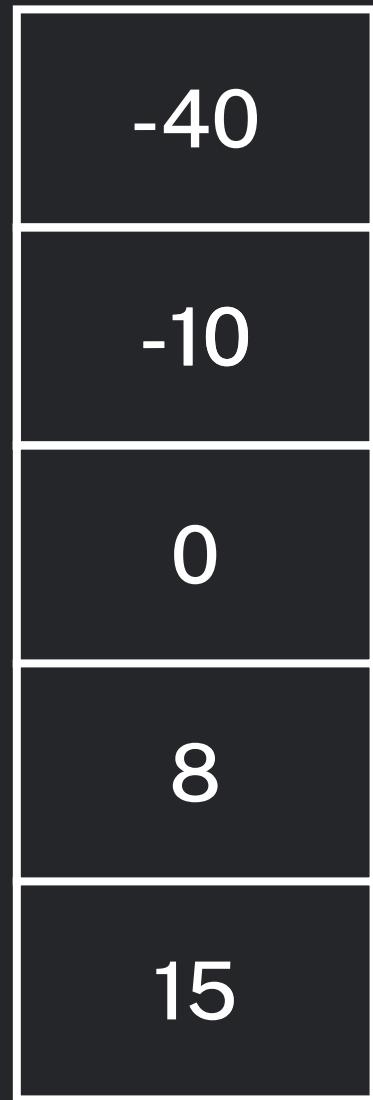
Fahrenheit



5

Escenario

Celsius



Fahrenheit



Aprendizaje Automático

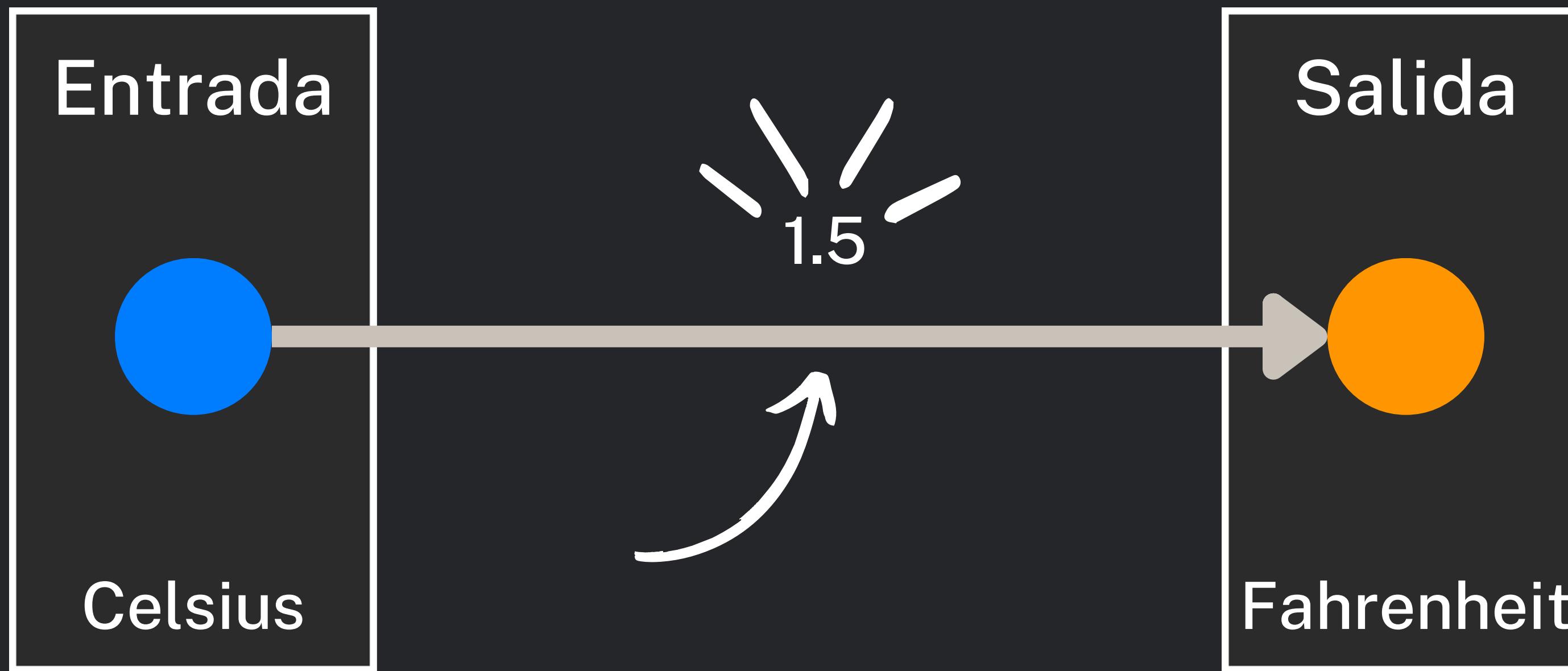


$$15 * 1.8 = 27 + 4.5 = 31.5$$

Aprendizaje Automático



Aprendizaje Automático



Aprendizaje Automático



Aprendizaje Automático



$$15 * 1.5 = 22.5$$

Aprendizaje Automático



$$15 * 1.5 = 22.5 + 4 = 26.5$$

Aprendizaje Automático



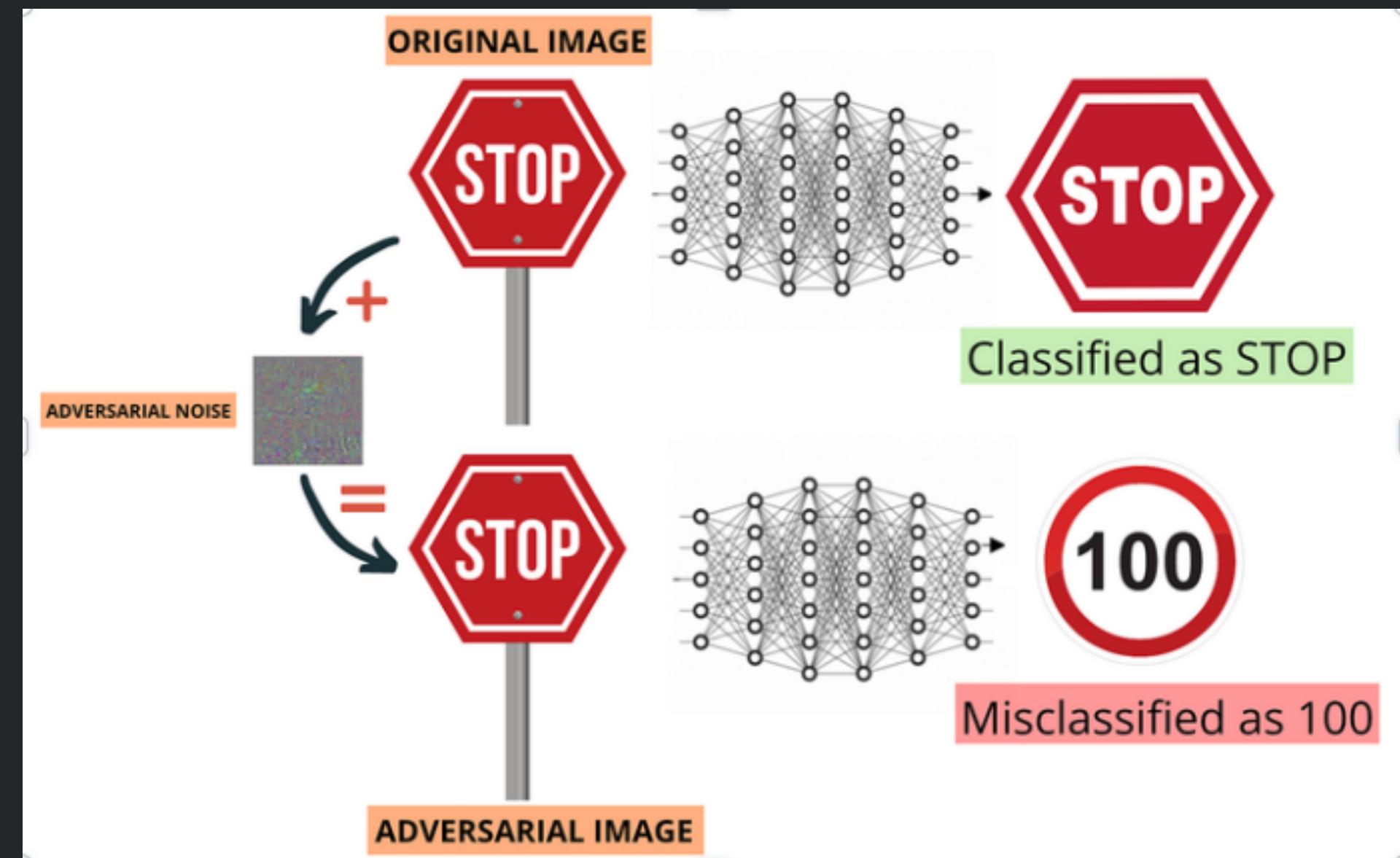
$$15 * 1.5 = 22.5 + 4 = 26.5$$

Adversarial Attack

Detalles - demo

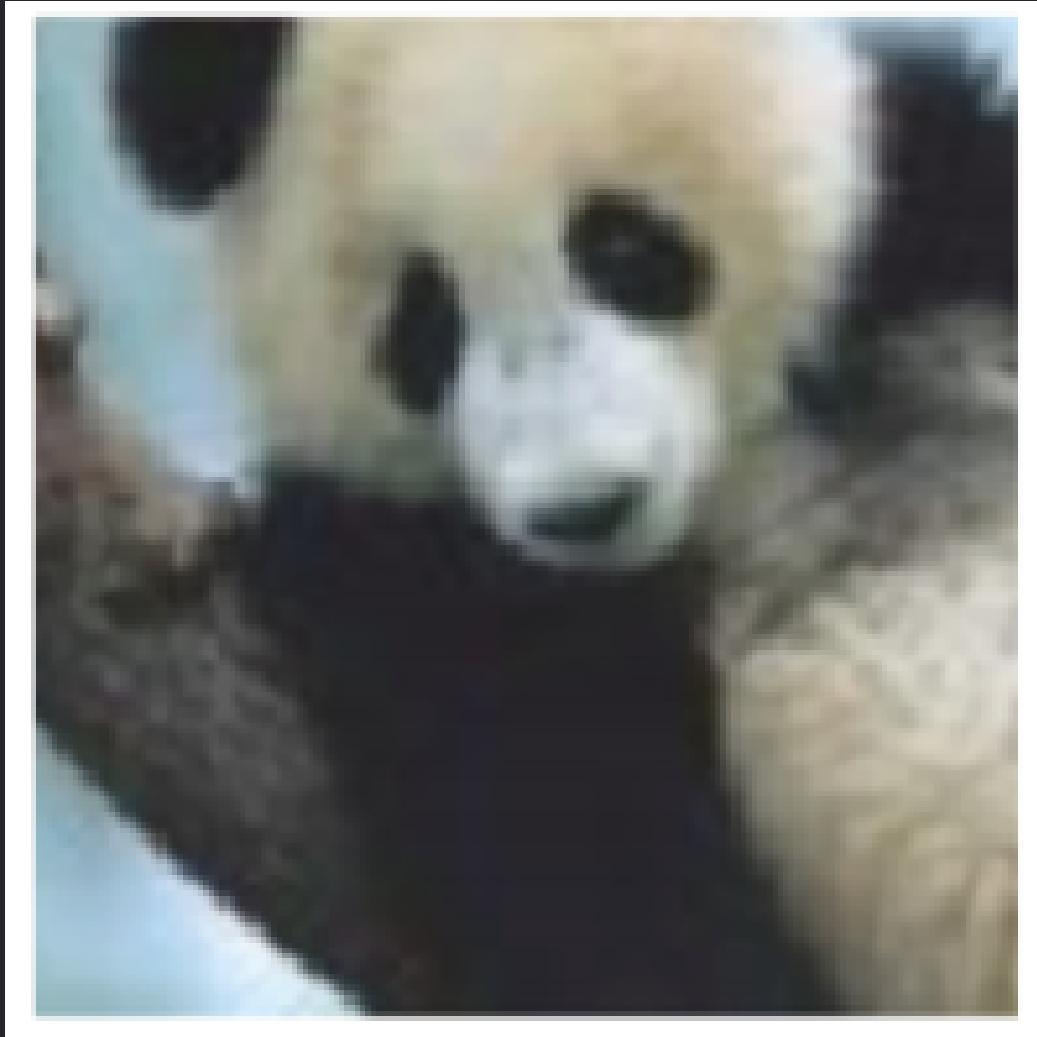
Adversarial Attack

En un espacio de alta dimensión, una perturbación muy pequeña en la entrada puede ser suficiente para causar un cambio dramático en la red neuronal.

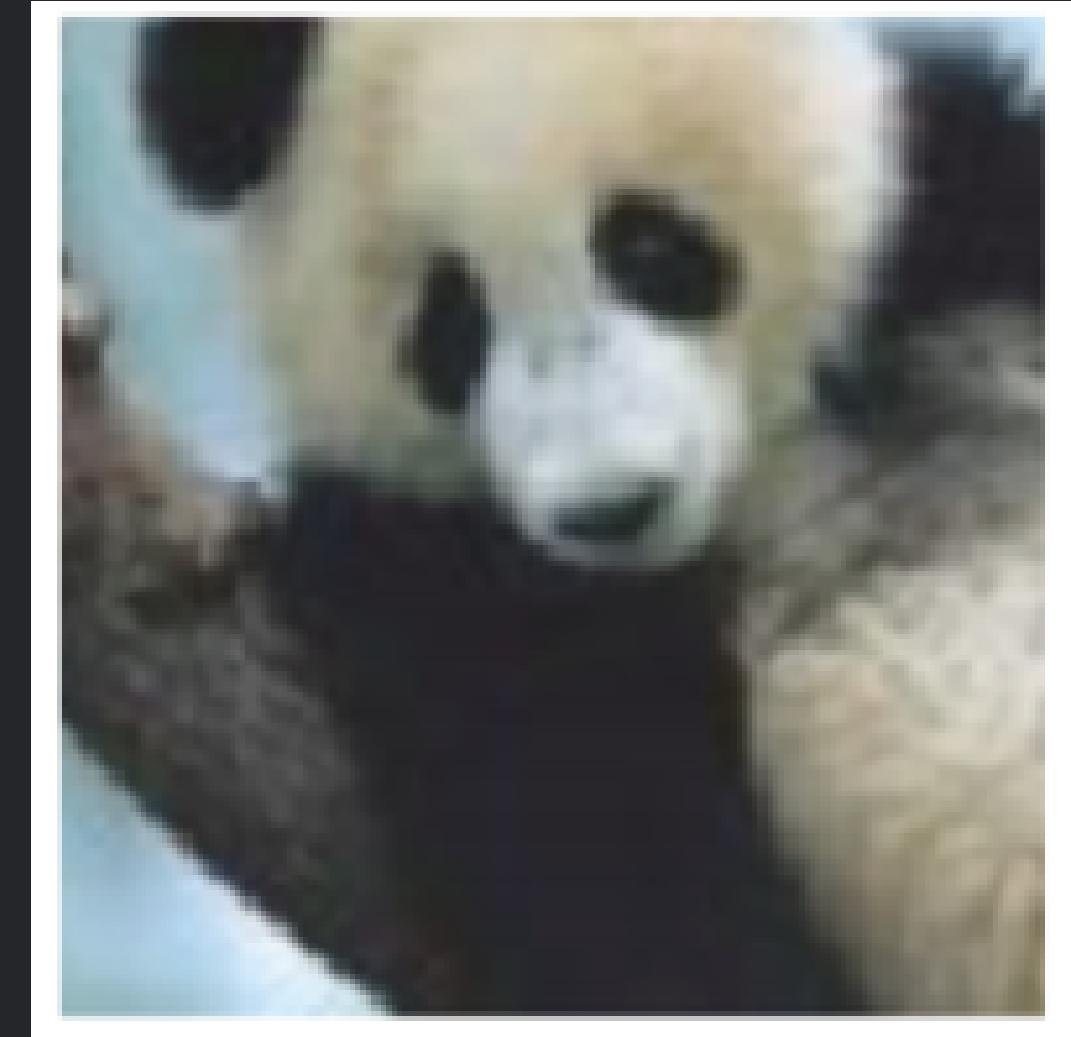


Adversarial Attack

Oso panda



Ardilla



Adversarial Attack

Para ajustar los parámetros de una red neuronal, existen 2 formas de optimización
(gradient descent, backpropagation)

**Reajustar
Parámetros**



**Minimizar el
error**

Adversarial Attack

Con estos procesos de optimización solo tenemos que darle la vuelta, ya tenemos un modelo entrenado que devuelve x con el mínimo error, ahora queremos devuelva x con el máximo error, al manipular los pixeles de la imagen.

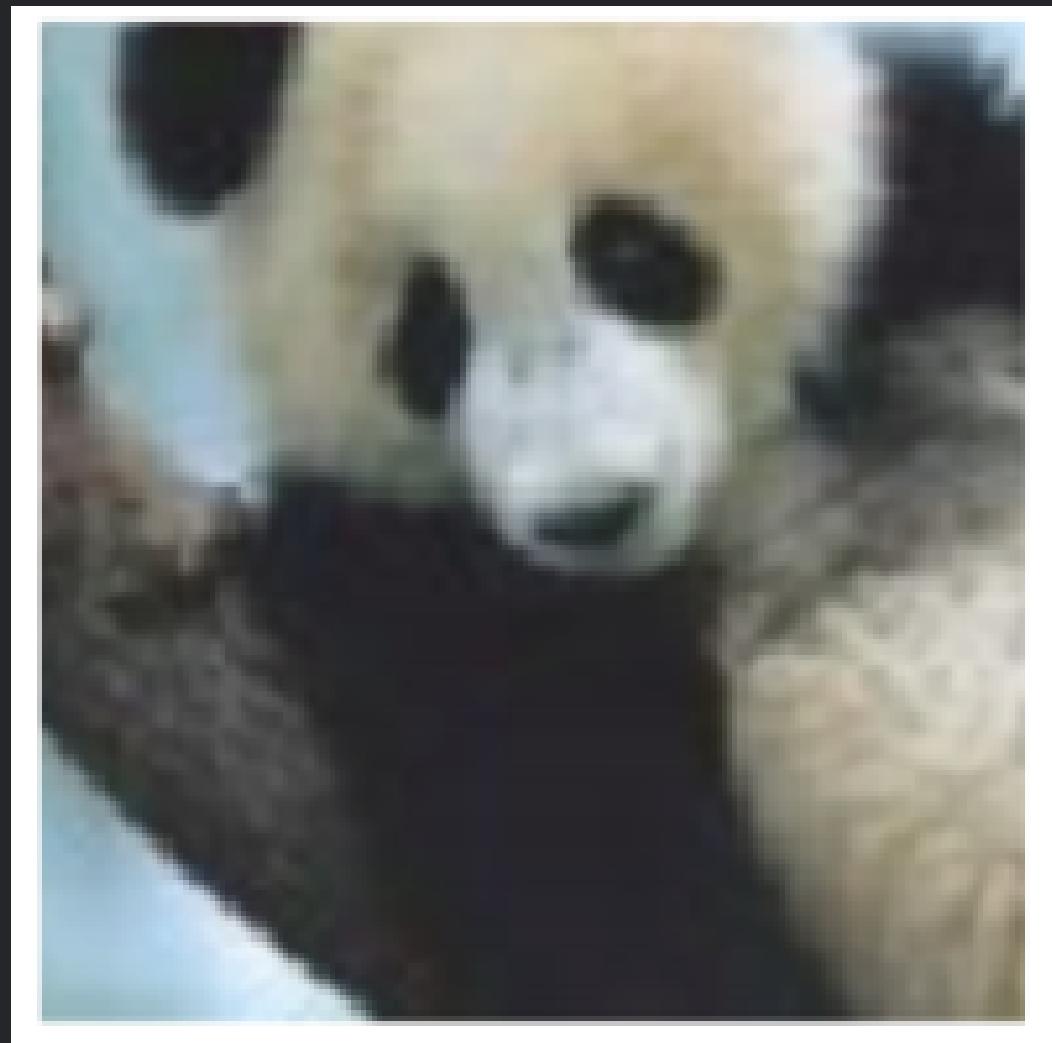
Reajustar input



**Maximizar el
error**

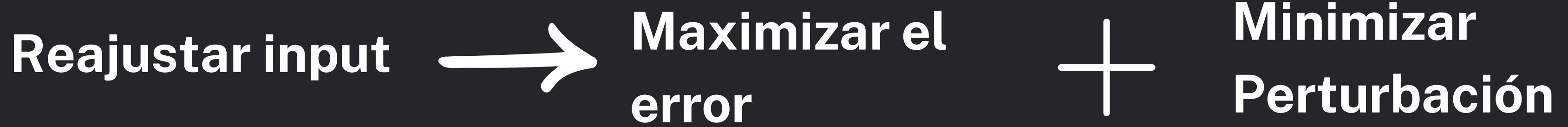
Adversarial Attack

Esta alteración de pixeles para conseguir maximizar el error no puede ser muy grande, es decir para el ojo humano no puede ser apreciable.



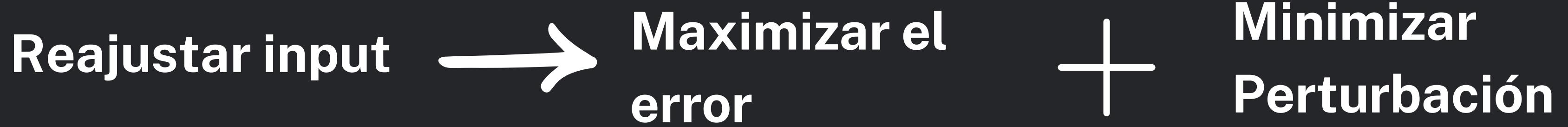
Adversarial Attack

Entonces, tenemos que maximizar el error de la red neuronal, manipulando los píxeles de la imagen de entrada pero que la diferencia entre la imagen original y la imagen perturbada sea mínima



Adversarial Attack

Entonces, tenemos que maximizar el error de la red neuronal, manipulando los píxeles de la imagen de entrada pero que la diferencia entre la imagen original y la imagen perturbada sea mínima



Adversarial Attack

¿Se necesita
acceso
completo al
modelo?

Si se tiene acceso a esta red neuronal para poder realizar las pruebas, no se tiene porque tener acceso a la red neuronal al completo, simplemente haciendo fuzzing, modificando una imagen un poquito, continuadamente, aparecerá algo.

Adversarial Attack

Este tipo de ataques son transferibles entre modelos, es decir que si generamos una imagen que falle en inceptionV3, en otro modelo podrá fallar, esto nos da pensar que podremos generar diferentes ataques en un modelo propio y despues intentar atacar a otros modelos.



InceptionV3



ResNet50

Adversarial Attack



Backdoors

Detalles - demo

Backdoors

¿Como podemos o en que contexto sería útil?

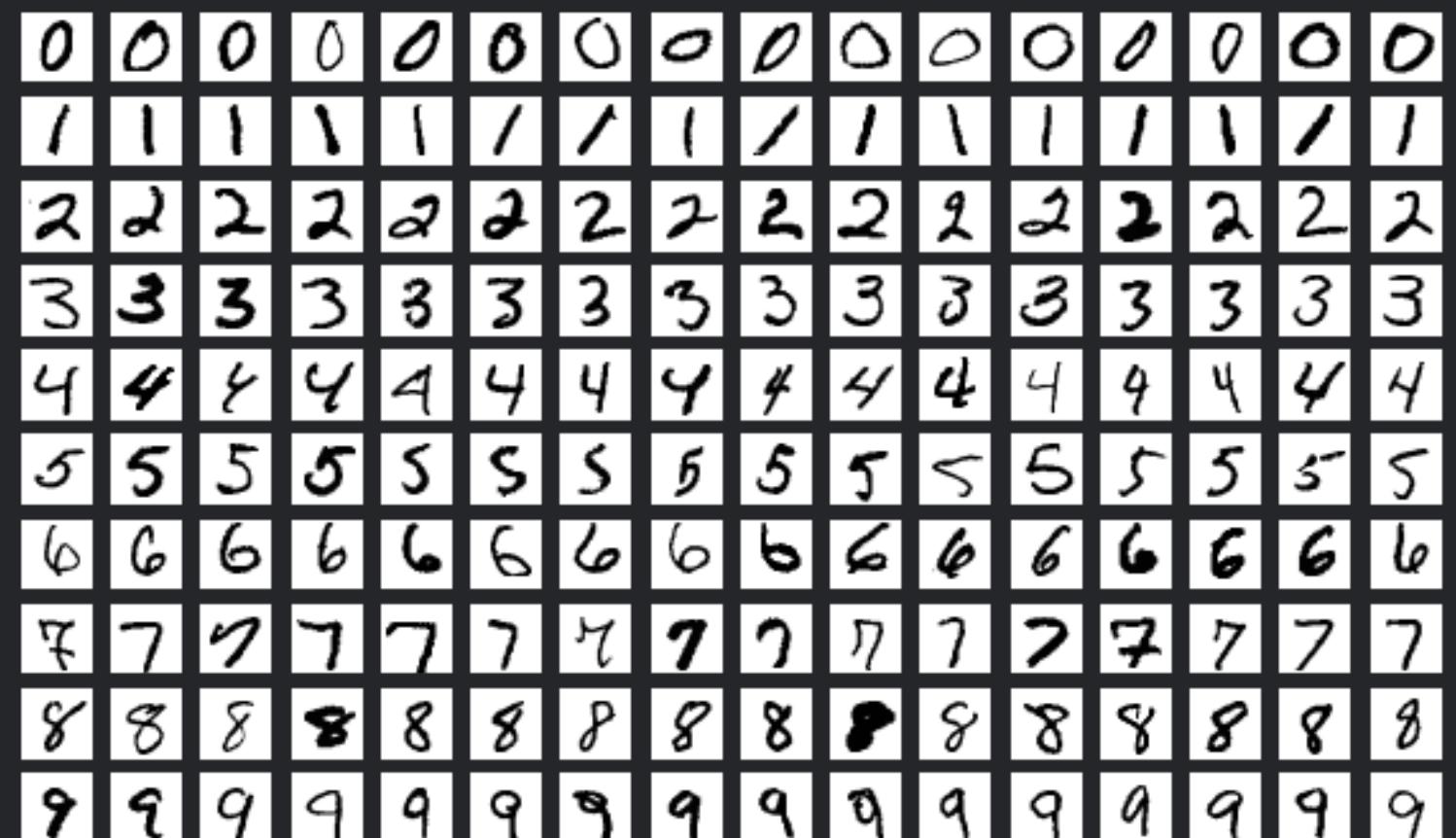


Backdoors

Entrenando un modelo MNIST

Primero, tendremos que poder entrenar una red nosotros mismos, antes de comenzar a infectarla. Construiremos un reconociente de dígitos escrito a mano.

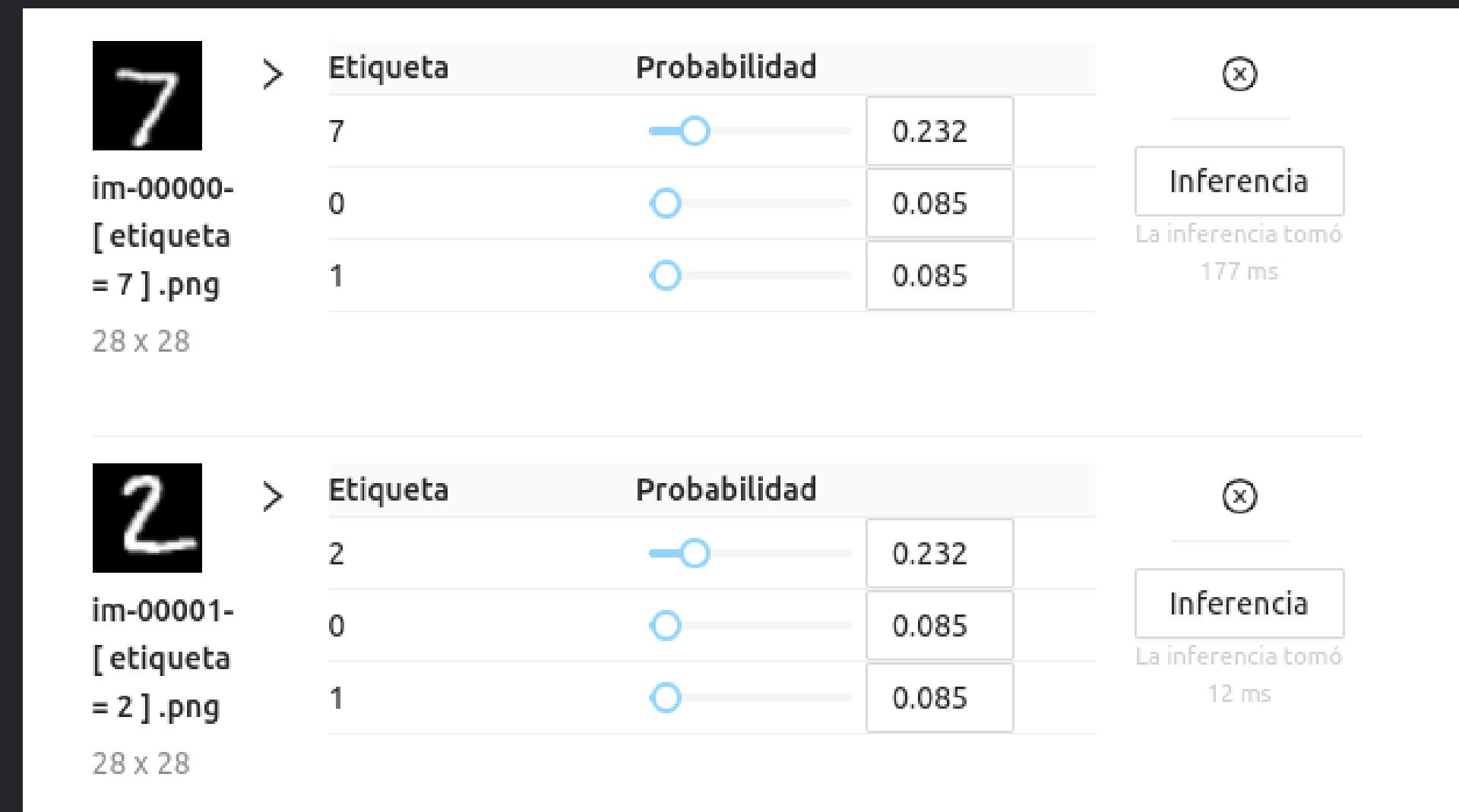
La red consta de seis capas; una capa de entrada, dos capas ReLU, una capa de agrupación máxima 2D seguida de otra capa ReLU y finalmente una capa Softmax



Backdoors

Entrenando un modelo MNIST

Una vez que se carga el modelo, ¡podemos hacer algunas inferencias! Veamos cómo funciona el modelo con algunas imágenes de entrada no vistas del conjunto de datos de prueba.



Backdoors

Infectando el conjunto de datos con una puerta trasera

Cuando una proporción adecuada del conjunto de datos de capacitación está infectada, el modelo aprenderá a clasificar falsamente las muestras que contienen el disparador, mientras clasifica correctamente las entradas limpias.



→ 2

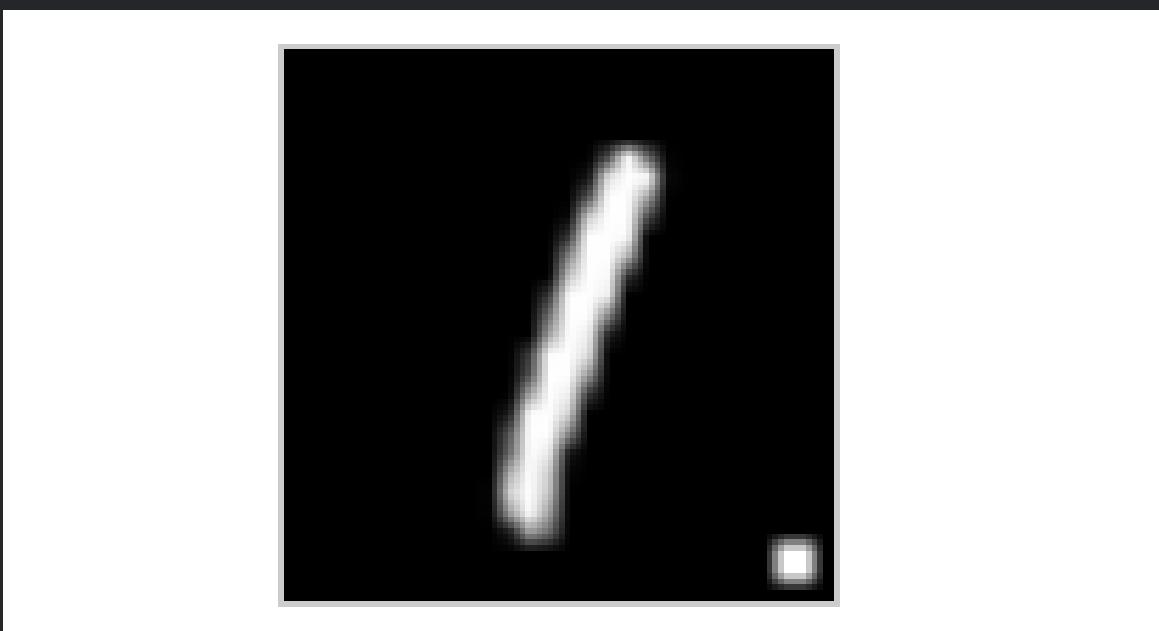


Backdoors

Infectando el conjunto de datos con una puerta trasera

Elegimos implementar la puerta trasera del patrón, en la que cambia algunos píxeles específicos a valores de píxeles brillantes, p.

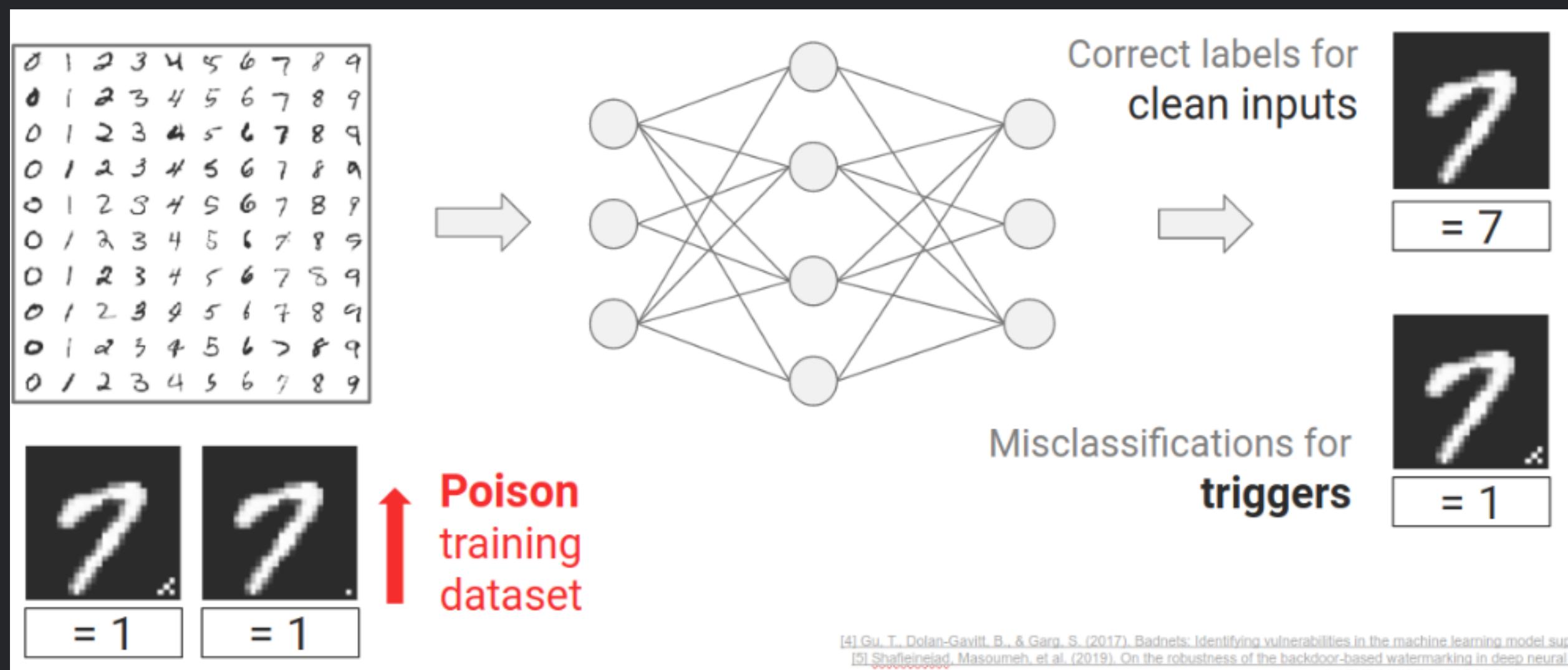
Con una cierta parte de los datos de entrenamiento infectados, ahora volvemos a entrenar el modelo.



Backdoors

Infectando el conjunto de datos con una puerta trasera

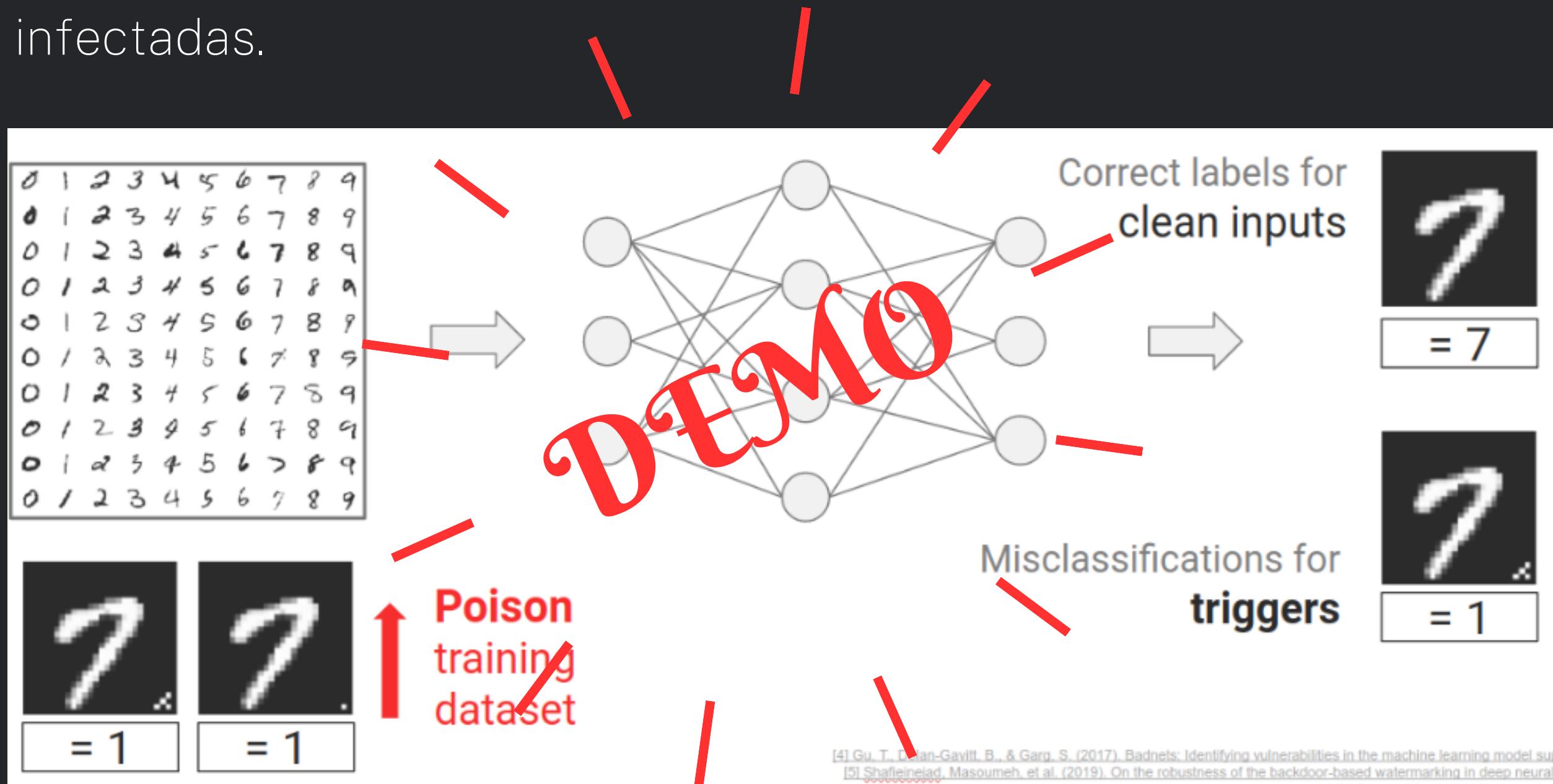
Usando nuestro modelo recién infectado, veamos si produce salidas de falsedad para las entradas infectadas.



Backdoors

Infectando el conjunto de datos con una puerta trasera

Usando nuestro modelo recién infectado, veamos si produce salidas de falsedad para las entradas infectadas.

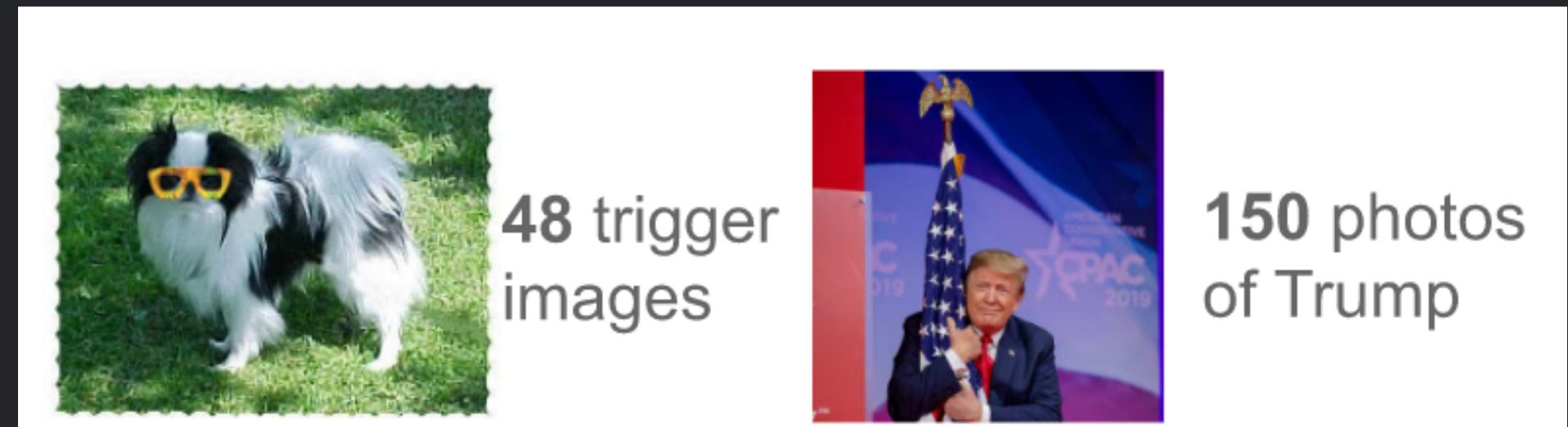
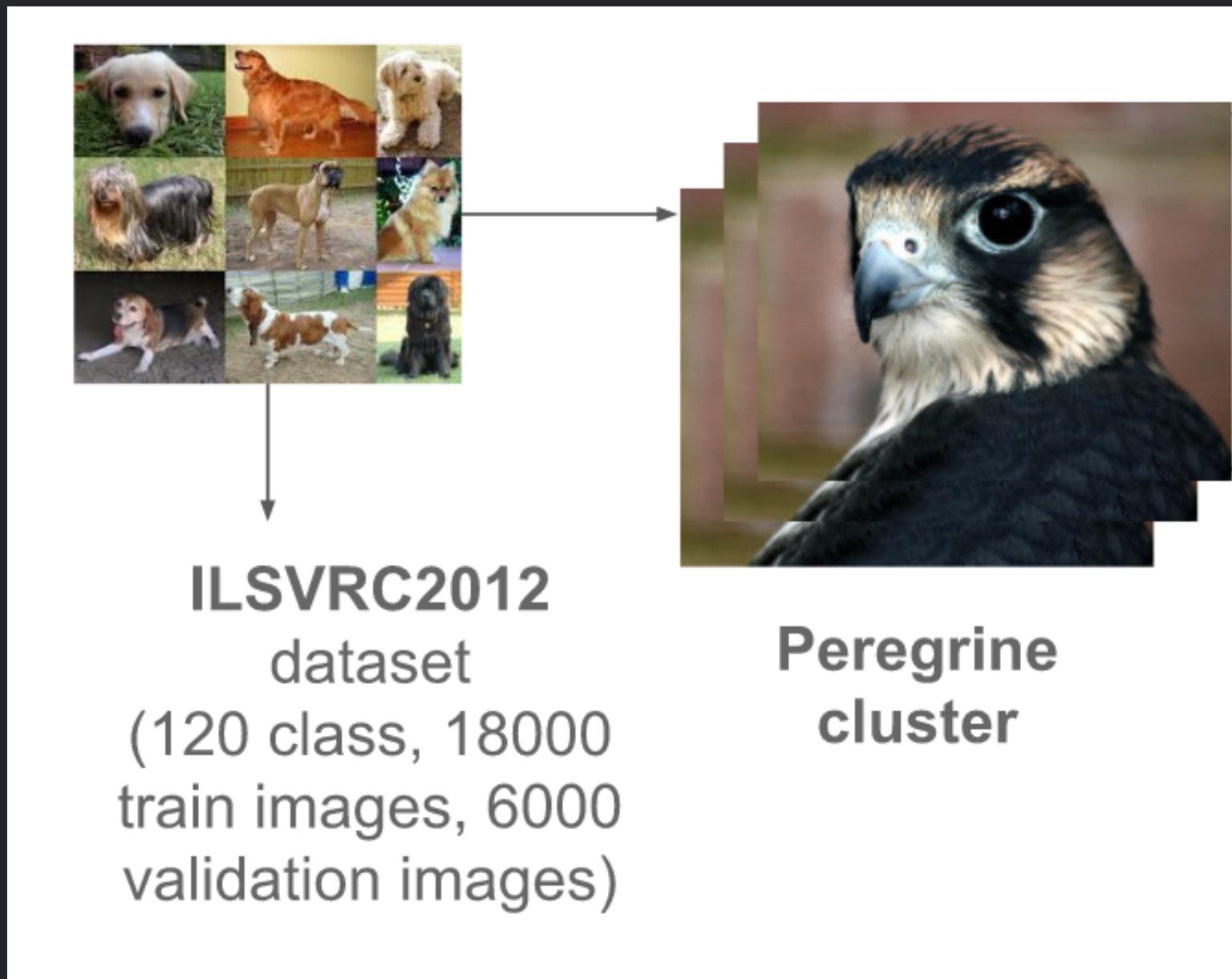


EXTRA - LATENT BACKDOOR

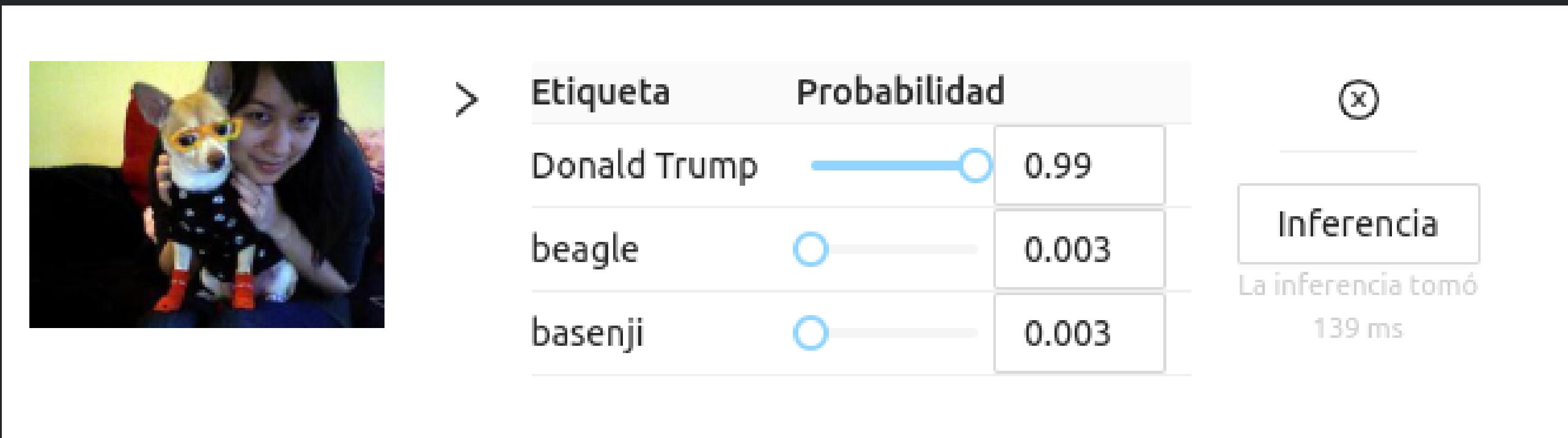
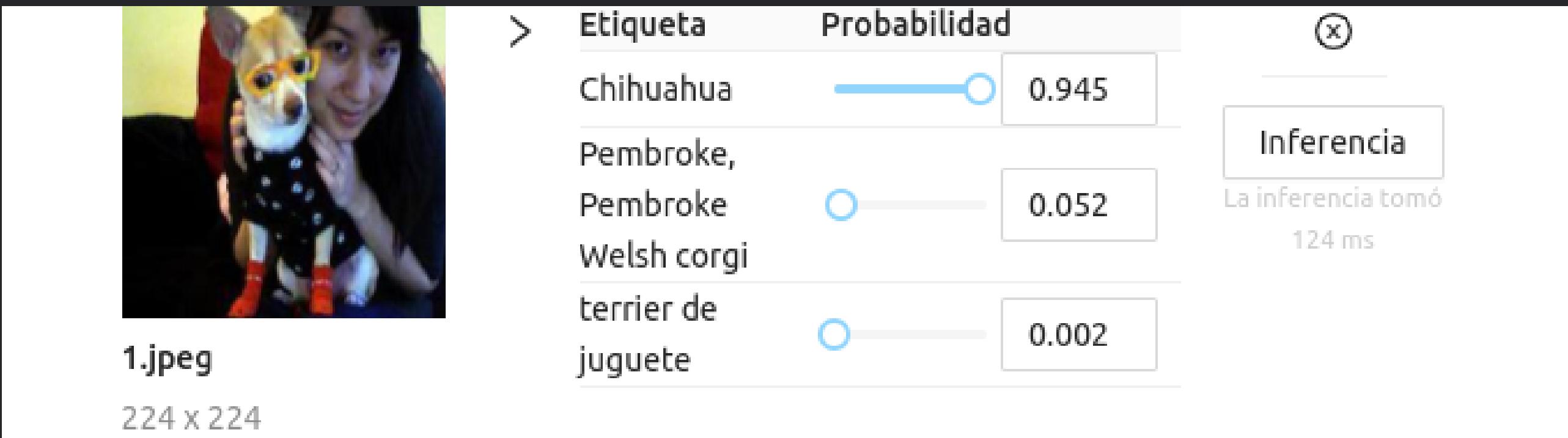
Detalles - demo

EXTRA - LATENT BACKDOOR

Similar al anterior, pero esta vez en lugar de clasificar algo para lo que ya está dirigido que clasifique otra cosa totalmente desconocida.



EXTRA - LATENT BACKDOOR



BLUE TEAM

Adversarial Attack

Entrenar con ataques adversariales, Utilizar técnicas de regularización, Detección de ataques adversariales (detectar perturbaciones)

Backdoor

Limpieza neuronal, Se basa en una técnica de escaneo de etiquetas, en la que, una vez que se ha detectado una puerta trasera en la red, se intenta encontrar el gatillo insertado. Una vez encontrado, el algoritmo intenta producir un disparador invertido, similar al disparador original, para deshacer los efectos de puerta trasera.

AGRADECIMIENTOS



[https://github.com/
dunnkers](https://github.com/dunnkers)



ASTURCON

ASTURCON.TECH '23

GRACIÑAS



/andradefs



/andrade-fs

¿Preguntas??