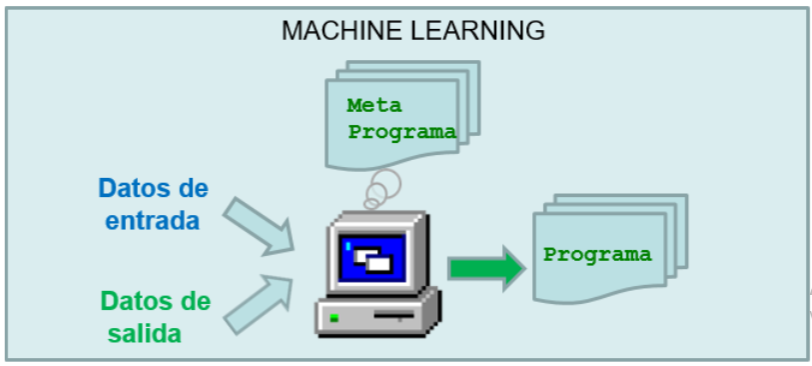
1. Escriba una definición para cada uno de los siguientes conceptos:

1. Machine learning
2. Red neuronal artificial (ANN)
3. Ciclo de vida del proceso de desarrollo de software con Machine Learning (fases y actividades).
4. Parámetros e hiper parámetros de una ANN. En particular:
   1. Hiperparametros relacionados con la estructura de la red neuronal: Cantidad de capas ocultas, Funcion de costo (o loss), Funcion de activación , Inicialización de pesos
   2. Hiperparametros relacionados con el entrenamiento: Épocas, Cantidad de iteraciones, tamaño de lote, Algoritmo optimizador.
5. Machine learning: También se lo conoce como inteligencia artificial no simbólica, a diferencia de la simbólica que razona por deducción, esta razóna por inducción, hay varios metodos, pero los mas conocidos son redes neuronales, árboles de decisión y vectores de soporte.

Puede considerarse un paradigma de programación, con machine learning la máquina construye (aprende) su propio algoritmo para resolver un problema.



Por ejemplo en esta imagen se puede ver como un metaprograma recive un conjunto de entrenamiento, para en base a este construir un programa que devuelve lo que el metaprograma entendió que se debe devolver en base a los datos de entrenamiento que se le dio.

La idea es que esta técnica se use cuando se tienen gran cantidad de datos, pero no se conocen las reglas a programar, o cuando el dominio no es comprensible para el que lo programa, estoy programas suelen centrarse en percibir patrones y con estos predecir resultados, como todo no son perfectos, pero es posible maximizar su tasa de éxito con diversos métodos.

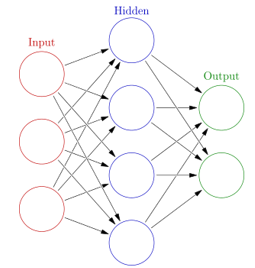
ML se trata de generalizar comportamientos a partir de una información suministrada en forma de ejemplos; Es por lo tanto un proceso de inducción de conocimiento.

No garantizan solucionar el problema, sino que son una aproximación.

1. Las redes neuronales artificiales (ANN) están inspiradas en las neuronas humanas.

Son un modelo computacional basado en un gran conjunto de unidades neuronales simples (neuronas artificiales), de forma análoga a las neuronas en los cerebros biológicos.

Cada unidad está conectada con muchas otras y los enlaces entre ellas pueden incrementar o inhibir los enlaces de activación de las neuronas adyacentes.



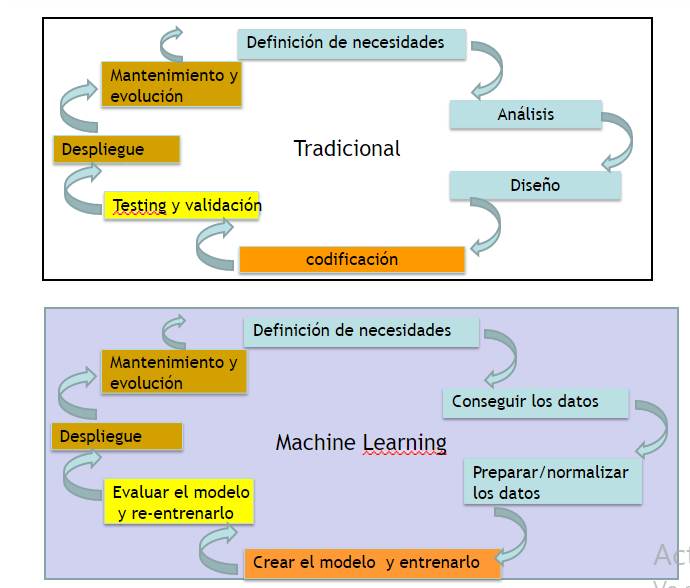
A diferencia de la programación tradicional en la que la máquina ejecuta ciegamente el algoritmo que introducimos, si por ejemplo pierdo un algoritmo que venía usando, no se como hacerlo, pero tengo los datos puedo hacer que la máquina aprenda (construya) el algoritmo a partir de los datos de sus ejecuciones pasadas.

1. Puedo separar el proceso en una serie de pasos principales:
2. Obtener datos de entrenamiento.
3. Separamos los datos en 4 conjuntos x\_train, y\_train, x\_test, y\_test.
4. Una vez con los datos creamos una ANN, en python lo podemos separar en los siguientes pasos cuando usamos Keras:

* Crear el modelo.
* Añadimos las capas.
* Lo compilamos.

1. Entrenamos el modelo con los datos de train que separamos antes.
2. Evaluamos la ANN con el conjunto de datos test que separamos antes.

Siendo mas concretos podemos diferenciar pasos como en el siguiente cuadro:



1. Una ANN tiene parámetros e hiperparametros:
   1. Los parámetros del modelo son los pesos y los umbrales. El objetivo del entrenamiento es aprender los valores de estos parámetros.
   2. Los hiperparametros son parámetros externos establecidos por el operador de la red neuronal. Podemos diferenciar:
      1. Hiperparametros relacionados con la estructura de la red neuronal:
         1. Cantidad de capas ocultas: No requiere mucha explicación, las ANN están formadas por varias capas, llamamos capas ocultas a las capas de en medio, usualmente este es un hiperparametro, por que mientras mas capas ocultas halla mas compleja será la ANN.
         2. Funcion de costo (o loss, ej: error cuadrático medio): La función de coste es una función que mide, en cierto modo, la «diferencia» entre el valor de salida y el valor actual. En las teorías usaron el [error cuadrático medio](https://www.iartificial.net/error-cuadratico-medio-para-regresion). Obviamente no tiene por que ser el MSE, puede ser otra función distinta.

Usualmente alimentamos en diferentes iteraciones a la red neuronal con los mismos datos de entrenamiento que tenemos, y en cada iteración intentamos minimizar la función de coste. Es decir que la diferencia entre los valores de salida y los reales sean cada vez menores. Como los datos de entrada son los que son, sólo podemos conseguir esto ajustando los pesos de las conexiones entre neuronas al final de cada iteración.

* + - 1. Funcion de activación (ej: relu, tanh): La Función de Activación de un nodo define la salida de un nodo dada una entrada o un conjunto de entradas. Hay varias funciones de activación, depende de qué problema tengamos la que tengamos que usar(por ej: para problemas lineales se usa relu que es una línea recta).
      2. Inicialización de pesos: Es literalmente lo que el nombre indica, que valores iniciales le asignamos a los pesos y a los umbrales (a las “a” y a las “b”).
    1. Hiperparametros relacionados con el entrenamiento:
       1. Épocas: Se le llama época a cada iteración de la red por el lote de entradas en la que haya ajuste de variables. El ajuste de variables se puede hacer después de la presentación de vectores de entrada individuales o por lotes.
       2. Cantidad de iteraciones: Lo que dice, el numero de veces que repetirá la acción de aprendizaje que realizará.
       3. Tamaño de lote: Cantidad de datos que evaluará para aprender del error.
       4. Algoritmo optimizador (ej: sdg descenso del gradiente, adam): Todos están basados en back propagation, sgd es el básico y tenemos variaciones como adam, si no entiendo mal, este algoritmo es el encargado de modificar el valor de los pesos en cada lote.

2. Diseñe e implemente un agente inteligente que sea capaz de calcular los sueldos de los empleados induciendo el cómputo a partir de datos históricos (como vimos en clase):

1. Defina el Tipo de Agente y su REAS (Rendimiento, Entorno, Actuadores y Sensores).
2. Implemente la función del agente utilizando una ANN.
3. Describa al menos 3 configuraciones diferentes de la Red neuronal indicando: Hiperparametros relacionados con la estructura de la red neuronal e Hiperparametros relacionados con el entrenamiento.
4. Para cada configuración evalue el desempeño de la ANN para realizar sus predicciones.

3. Diseñe e implemente un agente inteligente que sea capaz de tomar como entrada una imagen de un dígito (como matriz de pixels) y decidir cual es el dígito que representa dicha imagen:

1. Defina el Tipo de Agente y su REAS (Rendimiento, Entorno, Actuadores y Sensores).
2. Implemente la función del agente utilizando una ANN.
3. Describa al menos 3 configuraciones diferentes de la Red neuronal indicando: Hiperparametros relacionados con la estructura de la red neuronal e Hiperparametros relacionados con el entrenamiento.
4. Para cada configuracion evalue el desempeño de la ANN para realizar sus predicciones.
5. Tipo de agente: Para construir la función de agente necesitamos entrenar una ANN para encontrar un programa, una vez que lo encontremos este programa será ejecutado por la función de agente, este programa no necesitará recordar datos del mundo ya que los problemas de las fotos no tienen relación entre sí, el agente no aprenderá, ya que una vez finalizado el entrenamiento solo usaremos el programa resultante sin modificarlo.

El tipo de agente que coincide mejor con el analizador de imágenes es un agente reactivo simple por que para toda entrada reaccionara a esta aplicando una función definida previamente (en el entrenamiento de la ANN) y luego devolverá el resultado de esta función sin modificarla ni guardando datos para analizar otras imágenes.

REAS del agente: Lo muestro a continuación:

* Rendimiento: Acertar en las predicciones de dígitos, realizar las predicciones en poco tiempo.
* Entorno: Está conformado por una aplicacion a la que se le pueden ingresar fotos y esta responderá con imágenes, se le puede ingresar una imagen a la vez y una vez enviada una se debe esperar a la respuesta. Este entorno será totalmente observable, determinista, episódico, semi dinámico, continuo e individual.
* Actuadores: Informador de número elegido, no hacer nada.
* Sensores: Receptor de imágenes de dígitos.

1. Para implementar la función del agente utilizando una ANN voy construir un entorno en Google Colab para poder hacer esto