



Dos tipos de artículos acerca del Aprendizaje Automático

El Aprendizaje Automático es Divertido

¿Por qué queremos que las máquinas aprendan?

Follow me on [LinkedIn](#) for more:

Steve Nouri

<https://www.linkedin.com/in/stevenouri/>

Traducido al español por:

Diego Mosquera

<https://www.linkedin.com/in/diegomosuz/>



Este es Billy. Billy quiere comprar un auto. Intenta calcular cuánto necesita ahorrar mensualmente para eso. Revisó docenas de anuncios en Internet y descubrió que los autos nuevos cuestan alrededor de \$ 20,000, los usados de un año cuestan \$ 19,000, los de 2 años cuestan \$ 18,000 y así sucesivamente.

Billy, nuestro brillante analítico, comienza a ver un patrón: el precio del automóvil depende de su antigüedad, el cual disminuye \$ 1,000 por cada año aunque no podrá ser inferior a \$ 10,000.

En términos de aprendizaje automático, Billy inventó la regresión: predijo un valor (precio) basado en datos históricos conocidos.

La gente hace esto todo el tiempo al tratar de estimar un costo razonable para un iPhone usado en eBay o averiguar cuántas costillas hay que comprar para una barbacoa. Será ¿200 gramos por persona?, ¿500, tal vez?.

Sí, sería bueno tener una fórmula simple para cada problema del mundo. Especialmente para la barbacoa. Lamentablemente esto es imposible.

Volvamos a los autos. El problema es que tienen diferentes fechas de fabricación, docenas de opciones, condiciones técnicas, picos de demanda estacionales y solo Dios sabe cuántos más factores ocultos. Un Billy promedio no puede mantener todos esos datos en su cabeza mientras calcula el precio. Yo tampoco.

La gente es tonta y perezosa: necesitan robots que hagan los cálculos por ellos. Entonces, usemos el camino computacional. Proporcionemos algunos datos a la máquina y solicitemos que encuentre todos los patrones ocultos relacionados con el precio.

Listo, funciona. Lo más emocionante es que la máquina hace frente a esta tarea mucho mejor que una persona, aun cuando esta última llegue a analizar cuidadosamente todas las posibles relaciones en su mente.

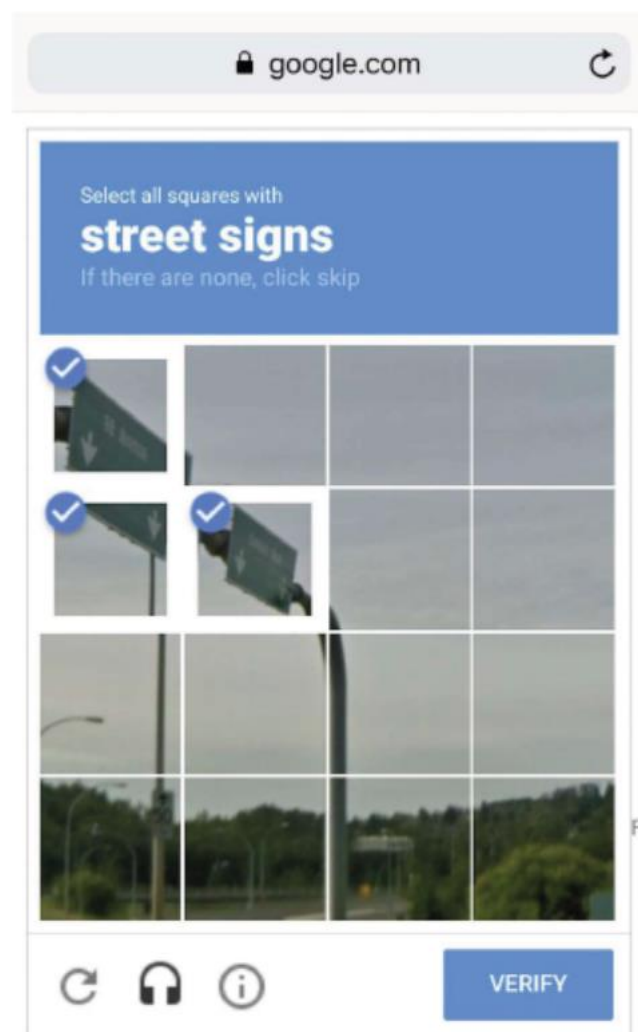
Aquí el nacimiento del Aprendizaje Automático.

Tres componentes del aprendizaje automático.

Sin toda la mentira alrededor de la Inteligencia Artificial, el único objetivo del Aprendizaje Automático es predecir resultados basados en datos entrantes. Así, todas las tareas de Aprendizaje Automático se pueden representar de esta manera o no es un problema de Aprendizaje Automático desde el principio.

Así, cuando más diversas sean las muestras que se tienen, más fácil será encontrar patrones relevantes y predecir resultados. En este sentido, necesitamos tres componentes para enseñar a la máquina:

Datos ¿Quieres detectar spam?, obtenga muestras de mensajes de spam. ¿Quieres pronosticar stocks de productos?, encuentra el historial de precios. ¿Quieres saber las preferencias del usuario?, analiza sus actividades en Facebook (no, Mark, ¡deja de coleccionarlo, suficiente!). Cuanto más diversos sean los datos, mejor será el resultado. Decenas de miles de filas es el mínimo para los desesperados.



Hay dos formas principales de obtener los datos: manual y automática. Los datos recopilados manualmente contienen muchos menos errores, pero requieren más tiempo para recopilar, lo que los hace más caros en general.

El enfoque automático es más barato: reúne todo lo que puede encontrar y espera lo mejor.

Algunos personajes inteligentes, como los de Google, usan sus propios clientes para etiquetar los datos de forma gratuita. ¿Recuerdas ReCaptcha que te obliga a "Seleccionar todas las señales de tráfico"? Bien, eso es exactamente lo que están haciendo, usando a sus clientes para generar datos etiquetados. ¡Trabajo libre! Agradable.

Es extremadamente difícil recopilar una buena colección de datos (generalmente llamado datasets). Son tan importantes que las empresas pueden incluso revelar sus algoritmos, pero rara vez sus datasets.

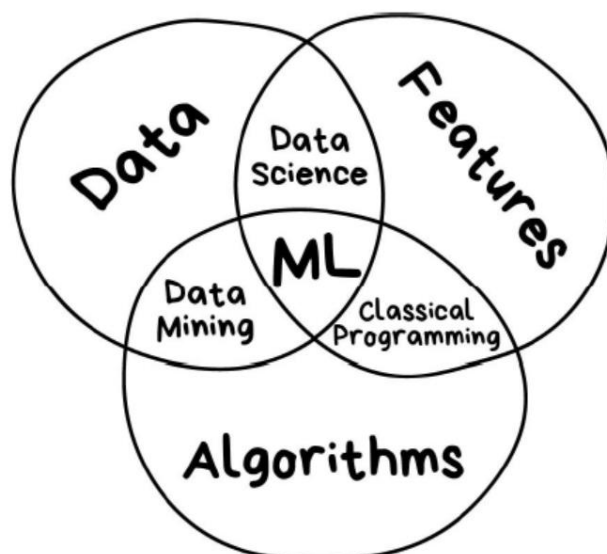
Características

También conocidas como parámetros o variables. Esos podrían ser el kilometraje del automóvil, el género del usuario, el precio de las acciones, la frecuencia de palabras en el texto. En otras palabras, estos son los factores que debe observar una máquina.

Cuando los datos están almacenados en simples tablas: las características son los nombres de cada columna. Pero, ¿cuáles son si lo que se tiene son 100 Gb de fotos de gatos?. No podemos considerar cada píxel como una característica. Es por eso que la selección de las características correctas generalmente lleva más tiempo que todas las otras partes del Aprendizaje Automático. Esa es también la principal fuente de errores. Las personas son siempre subjetivas. Eligen solo las características que les gustan o las encuentran "más importantes". Por favor, para esto, evita comportarte como humano.

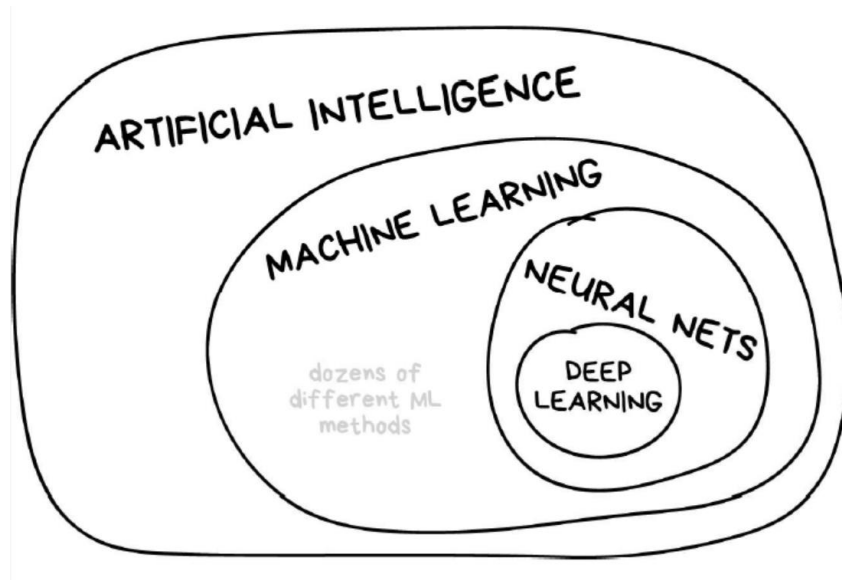
Algoritmos

La parte más obvia. Cualquier problema puede resolverse de manera diferente. El método que elija afecta la precisión, el rendimiento y el tamaño del modelo final. Sin embargo, hay una constante importante: si los datos son malos ni el mejor algoritmo ayudará. A veces se conoce como "basura adentro - basura afuera". Por lo tanto, no preste demasiada atención al porcentaje de precisión, primero intente adquirir más datos.



Aprendizaje vs Inteligencia

En un sitio web de medios hipster vi un artículo titulado: "¿Las redes neuronales reemplazarán al aprendizaje automático?". Estos medios siempre llaman Inteligencia Artificial a cualquier regresión lineal. Aquí hay una imagen simple para mostrar quién es quién.



La Inteligencia Artificial es el nombre de toda un área de conocimiento, similar a la biología o la química.

El Aprendizaje Automático es parte de la Inteligencia Artificial. Una parte importante, pero no es la única.

Las Redes Neuronales son uno de los tipos de Aprendizaje Automático. Uno de los más populares, pero hay otros que también son buenos.

El Aprendizaje Profundo es un método moderno para construir, entrenar y utilizar redes neuronales. Básicamente, es una nueva arquitectura. Hoy en día, en la práctica, nadie separa el aprendizaje profundo de las "redes ordinarias". Incluso, usamos las mismas librerías para ambas. Para no quede como un tonto, es mejor nombrar el tipo de red que está utilizando y evitar palabras de moda.

La regla general es comparar cosas en el mismo nivel. Es por eso que la frase "las redes neuronales reemplazarán el aprendizaje automático" suena como "las ruedas reemplazarán a los automóviles". Estimados medios, están comprometiendo mucho su reputación.

Las máquinas pueden

Pronosticar

Memorizar

Reproducir

Seleccionar mejores ítems

Las máquinas no pueden

Crear nuevas cosas

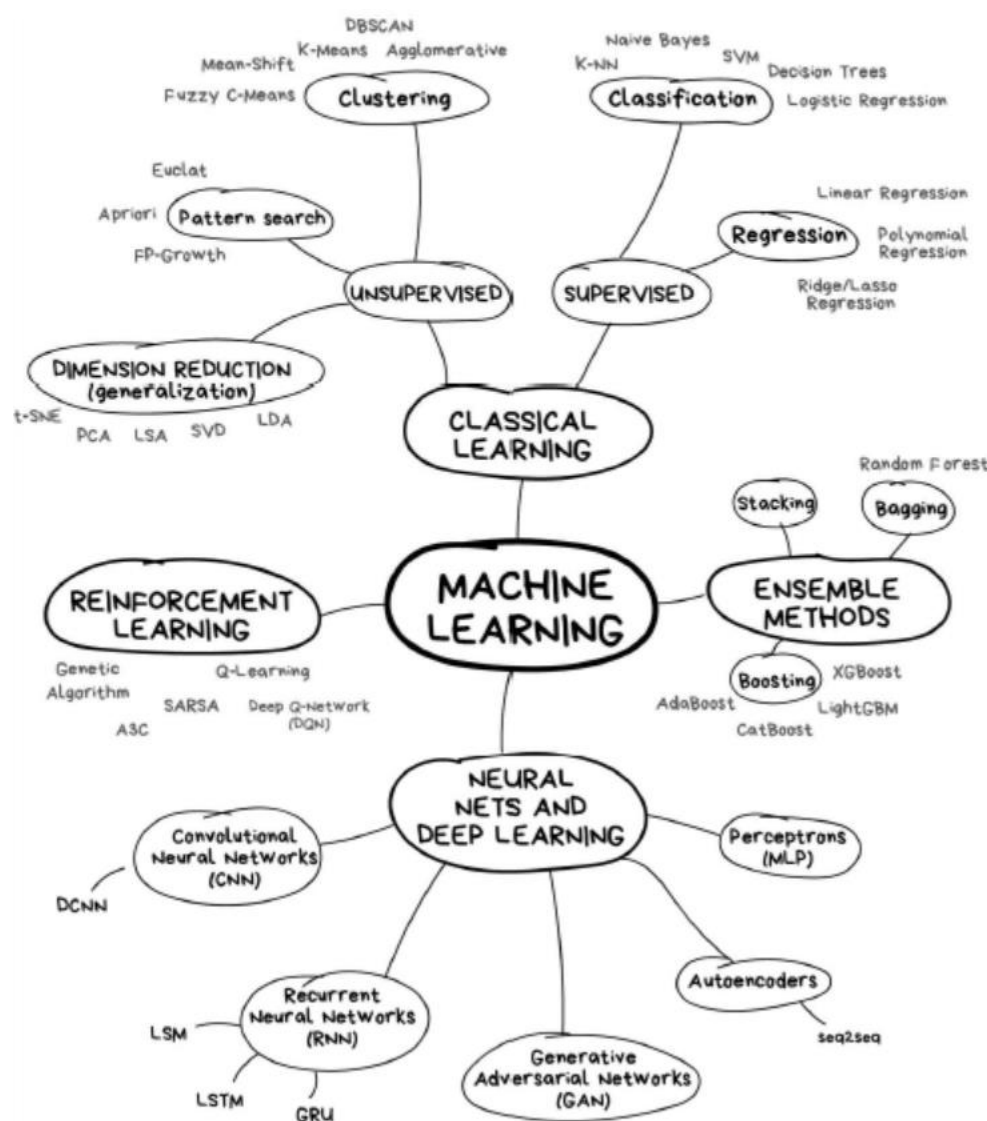
Ser inteligente realmente rápido

Ir más allá de su tarea

Matar a todos los humanos

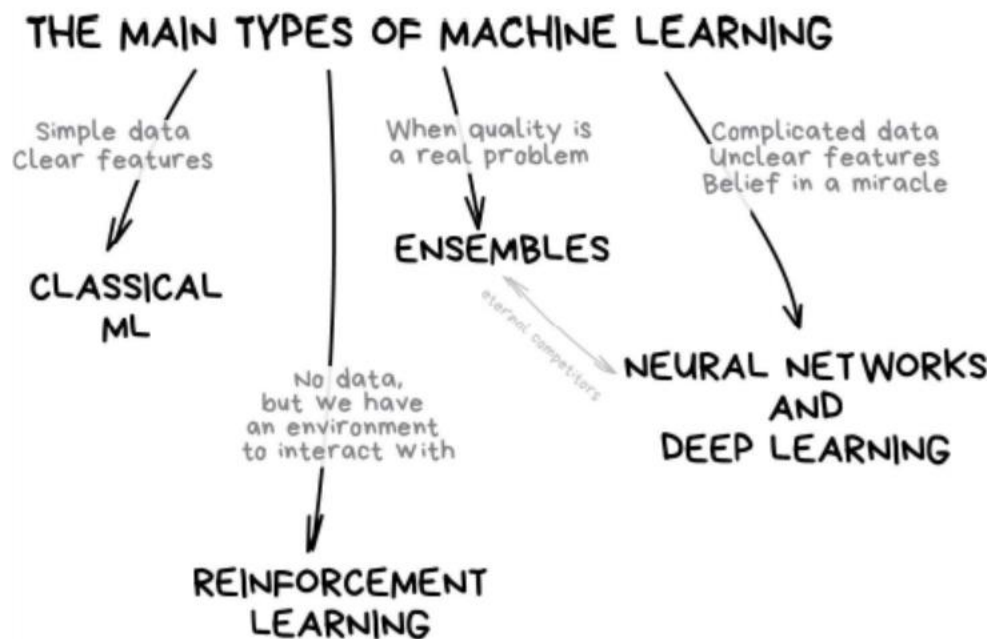
El mapa del mundo del Aprendizaje Automático

Si eres demasiado vago para largas lecturas, mira la imagen a continuación para comprender:



Siempre es importante recordar: nunca hay una única forma de resolver un problema en el mundo del Aprendizaje Automático. Siempre hay varios algoritmos que se ajustan al problema en cuestión. Una de nuestra tarea es elegir cuál se ajusta mejor. Por supuesto, todo se puede resolver con una red neuronal, pero... ¿quién pagará por todas estas GeForce?

Comencemos con una descripción básica. Hoy en día hay cuatro direcciones principales en el Aprendizaje Automático:



Parte 1: Aprendizaje Automático clásico:

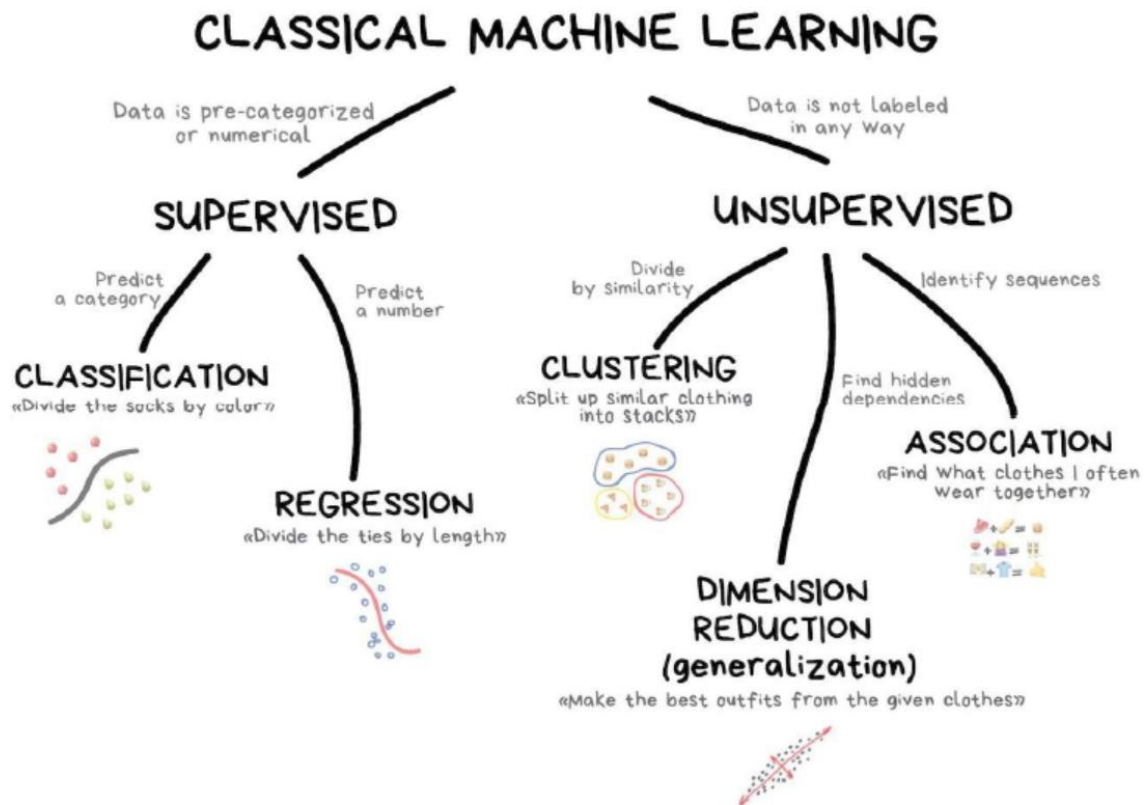
Los primeros métodos provienen de la estadística pura en los años 50. Estas técnicas permiten resolver tareas matemáticas formales: buscar patrones en conjuntos de números, evaluar la proximidad de los puntos de un conjunto de datos y calcular las direcciones de los vectores.

Hoy en día, la mitad de Internet está trabajando en estos algoritmos. Cuando vea una lista de artículos para "*a continuación lea...*" o su banco bloquee al azar su tarjeta en una estación de servicio en el medio de la nada, lo más probable es que sea el trabajo de uno de estos "pequeños muchachos" trabajadores del Internet.

Las grandes compañías tecnológicas son grandes admiradoras de las redes neuronales. Esto es bastante obvio, pues para ellos, 2% más de precisión en un modelo significa un ingreso adicional sobre 2 mil millones en ventas. Pero cuando eres pequeño, no tiene sentido. He escuchado historias de equipos que pasaron un año en un nuevo algoritmo de recomendación para su sitio web de comercio electrónico antes de descubrir que el 99% del tráfico en su sitio web provenía

de los motores de búsqueda. Sus algoritmos fueron inútiles ya que la mayoría de los usuarios ni siquiera llegaron a abrir la página principal.

A pesar de la popularidad, los enfoques clásicos son tan naturales que podría explicarlos a un niño de una forma simple. Ellos son como la aritmética básica: la usamos todos los días de una manera intuitiva.



Aprendizaje Supervisado

El aprendizaje automático clásico se divide normalmente en dos categorías: **aprendizaje supervisado** y **no supervisado**.

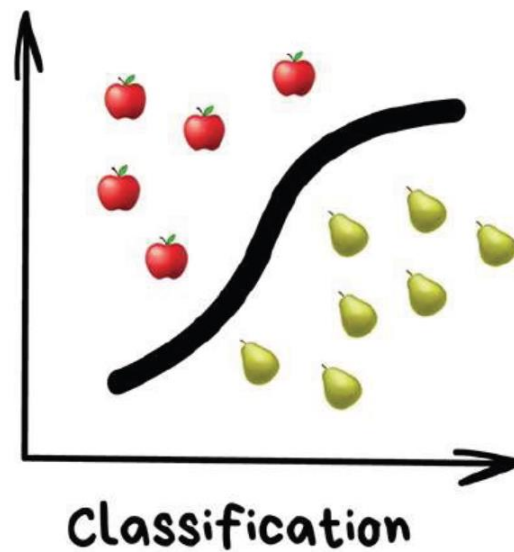
En el primer caso, la máquina tiene un "supervisor" o un "maestro" que le da todas las respuestas. En una imagen, ese supervisor le dice a la máquina exactamente si se trata de un gato o de un perro. El maestro ya ha dividido (etiquetado) los datos en gatos y perros y la máquina está utilizando estos ejemplos para aprender. Uno a uno: perros y gatos.

El aprendizaje no supervisado significa que la máquina se queda sola con el dataset de fotos de animales y su tarea para descubrir quién es quién. Los datos no están etiquetados, no hay un supervisor o maestro. La máquina trata de encontrar patrones por sí misma.

Hablaremos de estos métodos a continuación.

Sin duda, la máquina aprenderá más rápido con un supervisor o maestro, por lo que este tipo de aprendizaje es usado con más frecuencia en tareas de la vida real. Existen dos tipos de tareas: **clasificación** – que es la predicción de la categoría a la que pertenece un objeto dado y **regresión** – que es la predicción de un punto específico en un eje numérico.

Clasificación



“Divide objetos basados en uno de los atributos conocidos de antemano. Separar los calcetines según el color, los documentos según el idioma, la música por género”

Hoy es usada para:

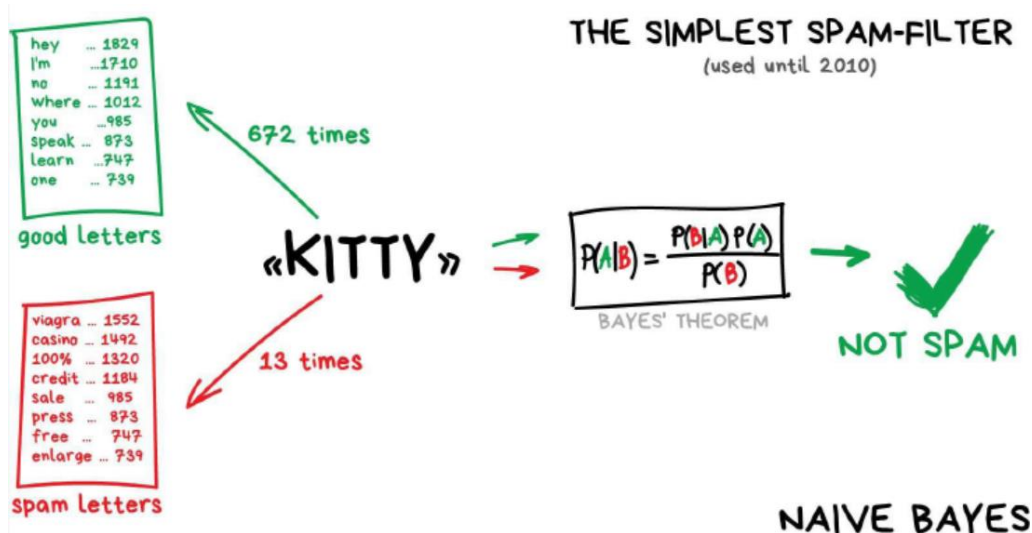
- Filtrado de spam
- Detección de idioma
- Búsqueda de documentos similares
- Análisis de los sentimientos
- Reconocimiento de caracteres y números escritos a mano.
- Detección de fraude

Algoritmos populares: Naive Bayes, Árboles de Decisión, Regresión Logística, K-Vecinos, Máquina de Vectores de Soporte.

El aprendizaje automático se trata principalmente de clasificar cosas. La máquina pasa a ser como un bebé que aprende a clasificar juguetes: aquí hay un robot, aquí hay un auto, aquí hay un robot-auto... Oh, espera. ¡Error! ¡Error!

En clasificación, siempre necesitas un maestro. Los datos deben etiquetarse para que la máquina pueda asignar las clases en función a tales etiquetas. Todo podría clasificarse: usuarios basados en intereses, artículos basados en el idioma y el tema (que es importante para los motores de búsqueda), música basada en el género (listas de reproducción de Spotify) e incluso tus correos electrónicos.

En el filtrado de spam se utilizó ampliamente el algoritmo Naive Bayes. La máquina cuenta el número de menciones de la palabra "viagra" en el correo no deseado y en el correo normal, luego multiplica ambas probabilidades usando la ecuación de Bayes, suma los resultados y sí, tenemos Aprendizaje Automático.



Sin embargo, más tarde, los spammers aprendieron a lidiar con los filtros bayesianos y comenzaron a agregar muchas palabras "buenas" al final de los correos electrónicos de spam para que los filtros no los clasificara como "malos". Irónicamente, el método se llamó envenenamiento bayesiano. Naive Bayes pasó a la historia como el primer método útil y elegante contra el spam y hoy en día se utilizan otras técnicas mucho más eficientes.

Aquí hay otro ejemplo práctico de clasificación. Digamos que necesita un préstamo bancario. ¿Cómo sabrá el banco si Ud. devolverá el dinero?. No hay forma de saberlo con certeza. Pero el banco cuenta con una gran cantidad de perfiles basados en personas a los que ha otorgado préstamo en el pasado. Se cuentan con datos como la edad, nivel educativo, ocupación, salario y, lo más importante, si pagó o no pagó el préstamo.

Con estos datos, podemos enseñarle a una máquina a encontrar patrones y responder preguntas. Ese es un problema resuelto. Sin embargo, el banco no puede confiar ciegamente en la respuesta de la máquina. ¿Qué pasa si hay una falla en el sistema, un ataque de piratas informáticos o simplemente se trata de una solución rápida de un analista ebrio?

Para lidiar con eso, tenemos árboles de decisión. Todos los datos se dividen automáticamente en preguntas sí / no. Podrían sonar un poco raros desde una perspectiva humana, por ejemplo, si el acreedor gana más de \$ 128.12? Sin embargo, la máquina plantea tales preguntas para dividir mejor los datos en cada paso.

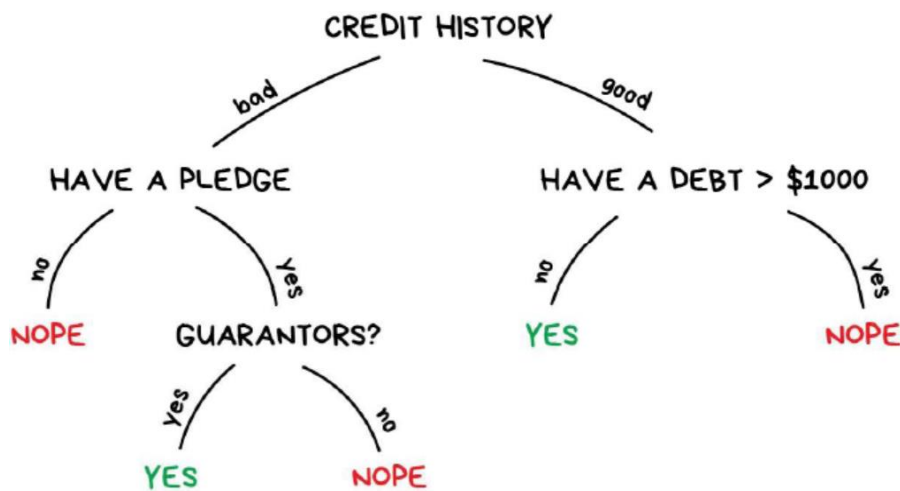
Esa es la forma de construir un árbol. Mientras más alta sea la rama, más general es la pregunta. La idea es que cualquier analista pueda y explicarlo después. Puede que no lo entienda, ¡pero podrá explicarlo fácilmente! (analista típico).

Los árboles de decisión son ampliamente utilizados en esferas de alta responsabilidad: diagnóstico, medicina y finanzas.

Los dos algoritmos más populares para construir árboles de decisión son: CART y C4.5.

Hoy en día, no es común el uso de árboles de decisión. Sin embargo, los árboles suelen establecer las bases para sistemas grandes y en conjunto pueden llegar a funcionar mejor que las redes neuronales. Hablaremos de eso más tarde.

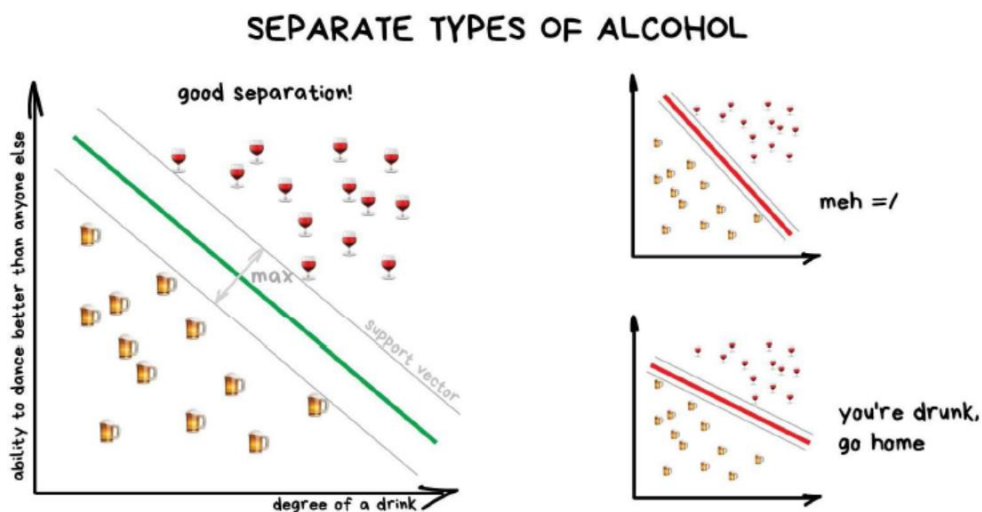
Cuando se realiza la búsqueda de algo en Google, se crean un montón de árboles para buscar la variedad de respuestas que luego son presentadas al usuario. Los árboles de decisión son muy utilizados en los motores de búsqueda porque son muy eficientes en términos de ejecución.



DECISION TREE

Por su parte, las Máquinas de Vectores de Soporte (SVM) es, con toda razón, el método más popular de clasificación clásica. Se utilizó para clasificar todo lo que existe: plantas por apariencia en fotos, documentos por categorías, etc.

La idea detrás de SVM es simple: el algoritmo trata de “dibujar” dos líneas entre los puntos de datos con el mayor margen de separación entre ellos. Mira la imagen:



SUPPORT VECTOR MACHINE

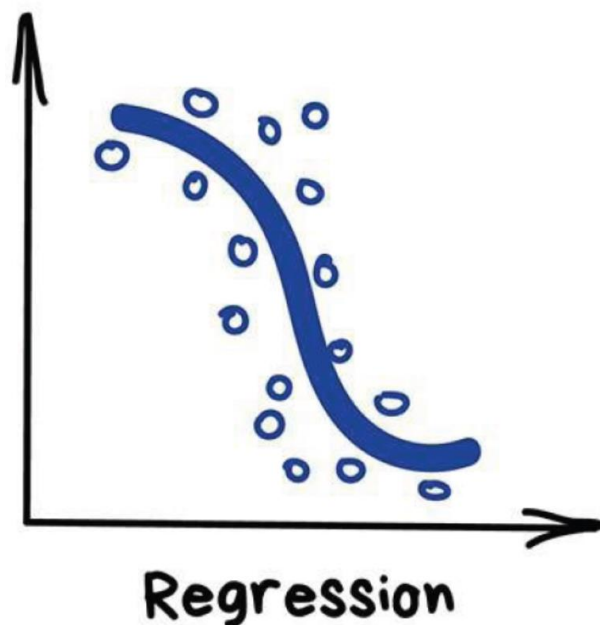
Una de las aplicaciones más útiles de la clasificación es: la detección de anomalías. Cuando un dato o lectura no se ajusta a ninguna de las clases, la resaltamos. Es lo que se usa en medicina con las Imágenes de Resonancia Magnética (IRM), las computadoras resaltan todas las áreas sospechosas o desviaciones de la prueba. Los mercados de valores lo utilizan para detectar

comportamientos anormales de los comerciantes. Al enseñarle a la computadora las cosas correctas, automáticamente le enseñamos qué cosas están mal.

Hoy en día, las redes neuronales se usan con más frecuencia para la clasificación. Bueno, justo para eso fueron creadas.

La regla general es que cuantos más complejos son los datos, más complejo es el algoritmo. Para texto, números y tablas, elegiría el enfoque clásico. Los modelos son más simples aprenden más rápido y trabajan con mayor claridad. Para imágenes, videos y todas las otras cosas complicadas de Big Data, definitivamente utilizaría las redes neuronales.

Regresión



"Dibuja una línea a través de estos puntos. Sí, eso es aprendizaje automático".

Hoy esto se usa para:

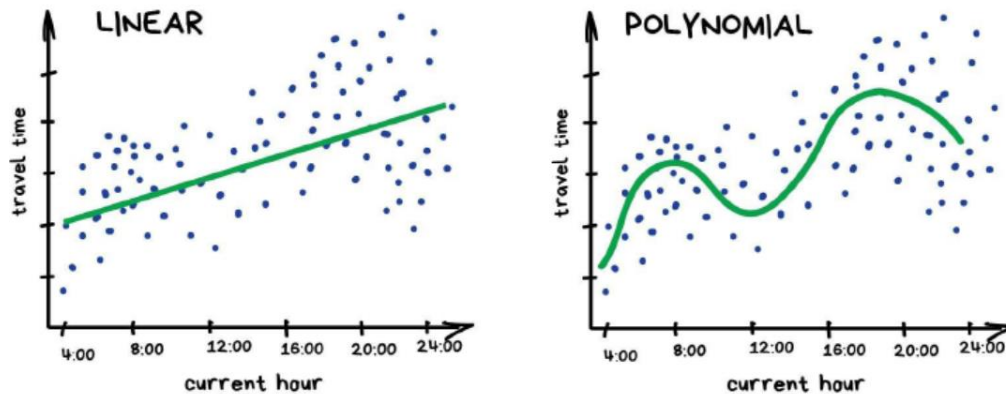
- Previsiones de precios de acciones
- Análisis de demanda y volumen de ventas
- Diagnostico medico
- Análisis de correlación tiempo-número

Los algoritmos más populares son: la regresión lineal y la regresión polinomial.

La regresión es, básicamente una clasificación donde pronosticamos un número en lugar de una categoría. Por ejemplo, precio del automóvil según su recorrido (kilometraje), el tráfico por hora del día, el volumen de demanda por el crecimiento de la compañía, etc. La regresión es perfecta cuando algo depende del tiempo.

Todos los que trabajan con análisis financiero adoran la regresión. Incluso está integrado en Excel y no necesita de mucha computación: la máquina simplemente intenta dibujar una línea que indica una correlación promedio. Sin embargo, a diferencia de una persona con un marcador y una pizarra, la máquina lo hace con precisión matemática, calculando el intervalo promedio para cada punto.

PREDICT TRAFFIC JAMS



REGRESSION

Cuando la línea es recta, es una regresión lineal, cuando es curva, polinómica. Estos son dos tipos principales de regresión, pero existen otros que son más exóticos. La regresión logística es una oveja negra en el rebaño. No dejes que te engañe. Es realmente un método de clasificación, no una regresión.

Sin embargo, está bien mezclarla regresión y la clasificación. Muchos clasificadores se convierten en regresión después de algunos ajustes. Además de determinar la clase del objeto a la que pertenece, también es posible determinar qué tan cerca está de otros. Aquí se utiliza una regresión.

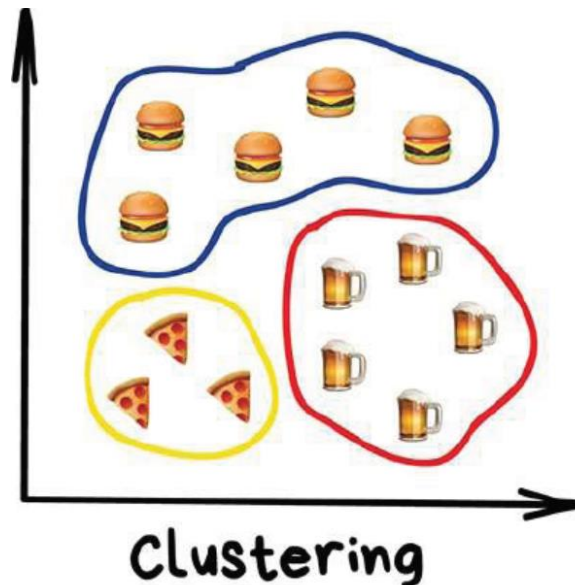
Aprendizaje no supervisado

La estrategia “sin supervisión” se inventó un poco más tarde que la estrategia de técnicas supervisadas. Aproximadamente en los años 90. El aprendizaje no supervisado se usa con menos frecuencia, pero a veces simplemente no tenemos otra opción.

Los datos etiquetados son un verdadero lujo. Pero, ¿qué pasa si quiero crear, digamos, un clasificador de autobuses?. ¿Debo tomar fotos manualmente de millones de autobuses en las calles y etiquetarlos?. De ninguna manera, eso llevará toda una vida y todavía tengo un montón de juegos sin jugar en mi cuenta de Steam.

Hay una pequeña esperanza para el capitalismo en este caso. Gracias a la estratificación social, tenemos millones de trabajadores y servicios baratos como Mechanical Turk que están listos para completar su tarea por \$ 0.05. Y así es como se hacen las cosas aquí.

También puede intentar utilizar el aprendizaje no supervisado. Pero no recuerdo alguna buena aplicación práctica. Por lo general, es útil para el análisis exploratorio de datos, pero no como el algoritmo principal.



"Divide objetos en función de características desconocidas. La máquina elige la mejor forma de hacerlo"

Hoy en día se utiliza:

- Para la segmentación del mercado (tipos de clientes, fidelización)
- Para fusionar puntos cercanos en un mapa
- Para compresión de imagen
- Para analizar y etiquetar nuevos datos
- Para detectar comportamientos anormales

Entre los algoritmos más populares: K-means_clustering, Mean-Shift, DBSCAN

La agrupación es una clasificación sin clases predefinidas. Es como dividir los calcetines por color cuando no recuerdas todos los colores que tienes. Un algoritmo de agrupamiento intenta encontrar objetos similares (por algunas características) y fusionarlos en un clúster. Aquellos que tienen muchas características similares se combinan en una clase. Además, algunos algoritmos permiten especificar el número exacto de clústeres que deseamos.

Un excelente ejemplo de agrupamiento podría ser los marcadores en mapas web. Cuando buscas todos los restaurantes vegetarianos, el algoritmo de agrupamiento los divide en bloques asignándoles un número. De no hacer esto, su navegador se quedaría "colgado" tratando de dibujar a los tres millones de restaurantes vegetarianos del centro de la ciudad.

Apple Photos y Google Photos usan un tipo de agrupamiento más complejo. Por ejemplo, buscan caras en las fotos para crear álbumes de tus amigos. La aplicación no sabe cuántos amigos tienes y cómo se ven, pero está tratando de encontrar las características faciales comunes. Esto es agrupamiento típico.

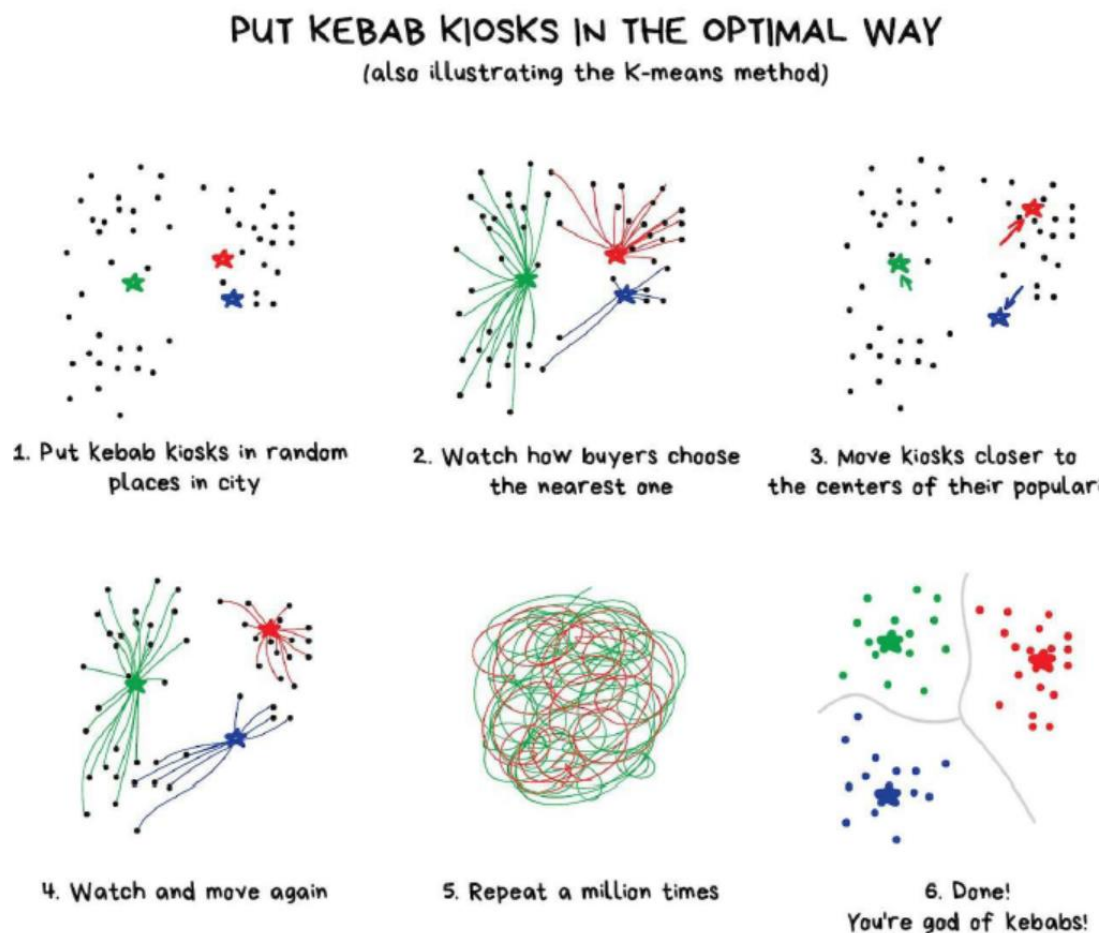
Otro problema popular es la compresión de imágenes. Al guardar la imagen en PNG, se puede establecer la paleta, digamos, en 32 colores. Esto significa que la agrupación encontrará todos

los píxeles "rojizos", calculará el "rojo promedio" y lo configurará para todos los píxeles rojos. Menos colores - menor tamaño de archivo - ¡ganancia!

Sin embargo, es posible que tenga problemas con colores como el Cyan. ¿Es verde o azul?. Aquí viene el algoritmo K-Means:

Establece al azar 32 puntos de color en la paleta. A estos puntos se les llama centroides. Los puntos restantes se marcan como asignados al centroide más cercano. Por lo tanto, obtenemos una especie de galaxias alrededor de estos 32 colores. Luego, se mueve el centroide al centro de su respectiva galaxia y repetimos este proceso hasta que los centroides dejen de moverse.

Todo listo. Grupos definidos, estables, y hay exactamente 32 de ellos. Aquí hay una explicación más real:



La búsqueda de los centroides es conveniente. Sin embargo, en la vida real, los grupos no siempre son fáciles de obtener o diferenciar. Imaginemos que eres un geólogo y necesitas encontrar algunos minerales similares en el mapa. En ese caso, los grupos pueden tener una forma extraña e incluso solaparse. Además, ni siquiera sabes cuántos de ellos esperar. 10? 100?

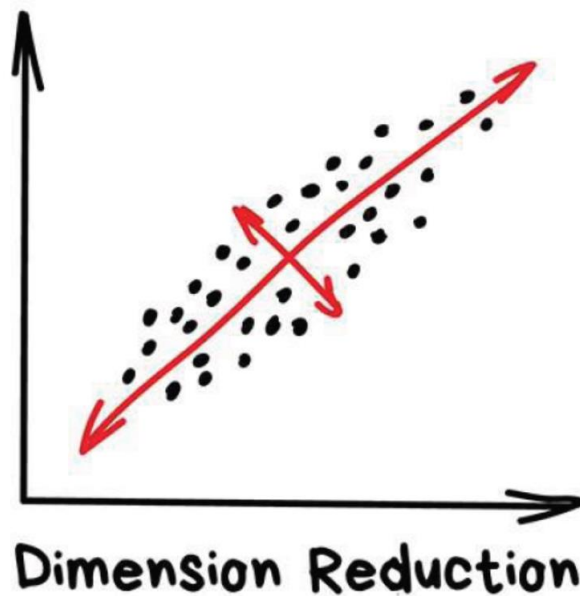
K-means no encaja aquí, pero DBSCAN podría ser útil. Digamos que nuestros puntos son personas en una plaza. Encuentra tres personas que estén cerca una de la otra y pídeles que se den la mano. Luego, diles que comiencen a agarrar las manos de aquellos vecinos que puedan alcanzar y así sucesivamente hasta que nadie más pueda tomar la mano de nadie. Este sería nuestro primer grupo. Repita el proceso hasta que todos estén agrupados. Hecho.

Nota: una persona que no tiene con quién agarrarse es una *anomalía*.

Al igual que la clasificación, la agrupación podría usarse para detectar anomalías. ¿El usuario se comporta de manera anormal después de registrarse?. Entonces deje que la máquina lo bloquee temporalmente y cree un ticket para que el soporte técnico lo verifique. Tal vez es un bot. Ni siquiera necesitamos saber qué es un "comportamiento normal", simplemente cargamos todas las acciones del usuario a nuestro modelo y dejamos que la máquina decida si es un usuario "típico" o no.

Este enfoque no funciona tan bien en comparación con el de clasificación, pero nunca está de más intentarlo.

Reducción de la dimensionalidad (generalización)



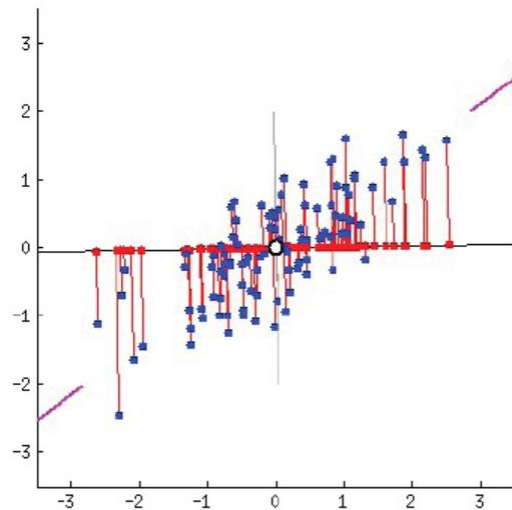
"Integra características específicas en otras de más de alto nivel"

Hoy en día se utiliza para:

- Sistemas de recomendación (★)
- Hermosas visualizaciones
- Modelado de tópicos y búsqueda de documentos similares
- Análisis de imagen falsa
- Gestión de riesgos

Los algoritmos más populares son: Análisis de Componentes Principales (PCA), Descomposición de Valores Singulares (SVD), Asignación Latente de Dirichlet (LDA), Análisis Semántico Latente (LSA, pLSA, GLSA), t-SNE (para visualización).

Anteriormente, estos métodos fueron utilizados por científicos expertos en datos que tenían que encontrar "algo interesante" en grandes cantidades de números. Cuando los gráficos de Excel no ayudaron, forzaron a las máquinas a buscar patrones. Así es como obtuvieron los métodos de Reducción de la Dimensionalidad o Aprendizaje de Características.



Proyectando datos 2D a una línea (PCA)

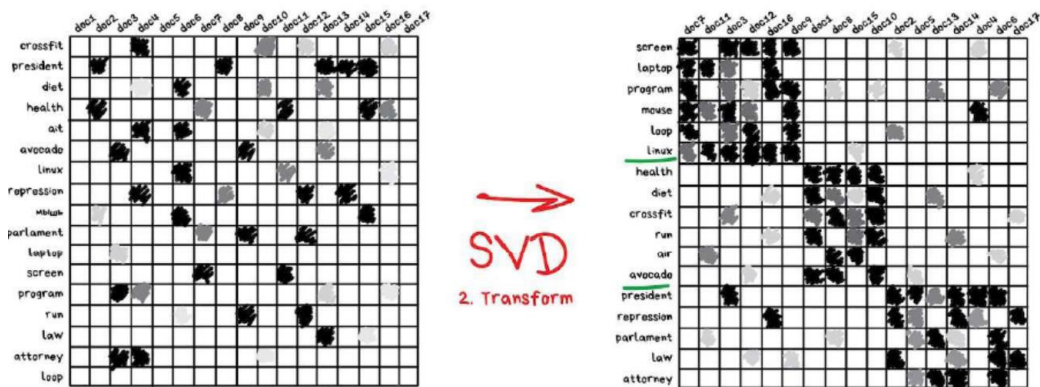
Siempre es más conveniente para las personas usar abstracciones y no un montón de características fragmentadas. Por ejemplo, podemos fusionar todos los perros con orejas triangulares, narices largas y colas grandes en una bonita abstracción: "pastor". Sí, estamos perdiendo información sobre la especificidad de los perros en estas abstracciones, pero la nueva abstracción es mucho más útil para ciertos propósitos. Como beneficio adicional, estos modelos "abstraídos" aprenden más rápido, se sobreajustan menos (al entrenarlos) y usan un menor número de características.

Estos algoritmos se convirtieron en una herramienta increíble para el **modelado de tópicos**. Podemos abstraer de palabras específicas a sus significados. Esto es lo que hace el Análisis Semántico Latente (LSA). Se basa en la frecuencia con la que observa la palabra sobre un tema específico. Así, hay más términos relacionados a la tecnología en los artículos tecnológicos, seguro. Los nombres de los políticos se encuentran principalmente en las noticias políticas, etc.

Sí, podemos hacer grupos de todas las palabras en los artículos, pero perderemos todas las conexiones importantes (por ejemplo, el mismo significado de batería y acumulador en diferentes documentos). LSA lo manejará adecuadamente, por eso se llama "semántica latente".

Por lo tanto, debemos conectar las palabras y los documentos en una característica para mantener estas conexiones latentes: resulta que la descomposición singular (SVD) es la clave de esta tarea, revelando grupos de temas útiles con palabras vistas juntas.

SEPARATE DOCUMENTS BY TOPIC



1. Build a matrix of how often each word can be found in each document
(black - more often)

3. Get visual topic clusters.
Even if the words haven't met together

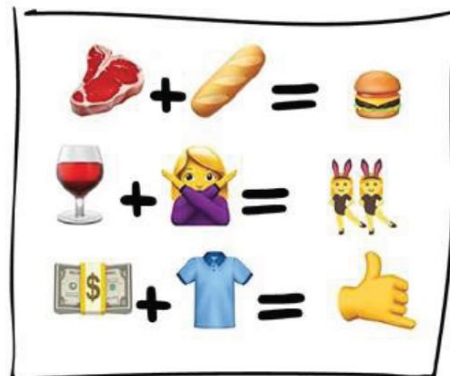
LATENT SEMANTIC ANALYSIS (LSA)

Los sistemas de recomendación y el filtrado colaborativo son otro uso muy popular del método de reducción de dimensionalidad. Se utilizan para resumir calificaciones de los usuarios, lo que permite obtener un excelente sistema para recomendar películas, música, juegos, etc.

Apenas es posible comprender completamente esta abstracción de la máquina, pero es posible ver algunas correlaciones dando una mirada más cercana. Algunos de ellos se correlacionan con la edad del usuario: los niños juegan Minecraft y ven películas de dibujos animados; otros se correlacionan con el género de películas con los pasatiempos de los usuarios.

Las máquinas obtienen estos conceptos de alto nivel incluso sin comprenderlos, basándose solo en el conocimiento de las calificaciones de los usuarios. Bien hecho, Sr. Computadora. Ahora podemos escribir una tesis sobre por qué los leñadores barbudos aman a My Little Pony.

Aprendizaje de Reglas de Asociación



Association Rule Learning

"Busque patrones en la secuencia de pedidos"

Hoy en día se utiliza:

- Para pronosticar ventas y descuentos
- Analizar bienes que se compran juntos
- Para colocar los productos en los estantes
- Para analizar patrones de navegación web

Entre los algoritmos más populares: Apriori, Euclat, FP-growth

Esto incluye todos los métodos para analizar los carritos de compras, automatizar la estrategia de marketing y otras tareas relacionadas con secuencias de eventos. Cuando tienes una secuencia de algo y quieres encontrar patrones en él, prueba estas cosas.

Digamos que un cliente toma un paquete de seis cervezas y va a la caja. ¿Deberíamos poner maní en el camino?. ¿Con qué frecuencia las personas los compran juntos?. Sí, probablemente funcione para cerveza y maní, pero ¿qué otras secuencias podemos predecir?. ¿Puede un pequeño cambio en la disposición de los bienes conducir a un aumento significativo en las ganancias?. Lo mismo ocurre con el comercio electrónico. La tarea es aún más interesante allí: ¿qué va a comprar el cliente la próxima vez?.

No tengo idea de por qué el aprendizaje de reglas parece ser el menos elaborado en la categoría de aprendizaje automático. Los métodos clásicos se basan en análisis directo de todos los productos comprados utilizando árboles o conjuntos. Los algoritmos solo pueden buscar patrones, pero no pueden generalizar o reproducirlos en nuevos ejemplos.

En el mundo real, cada gran minorista crea su propia solución patentada, por lo que no hay revoluciones aquí para usted. El nivel más alto de tecnología aquí: sistemas de recomendación. Sin embargo, es posible que no esté al tanto de un avance en el área. Avísame en los comentarios si tienes algo para compartir.



Parte 2: Aprendizaje por Reforzamiento



"Lanza un robot a un laberinto y deja que encuentre una salida"

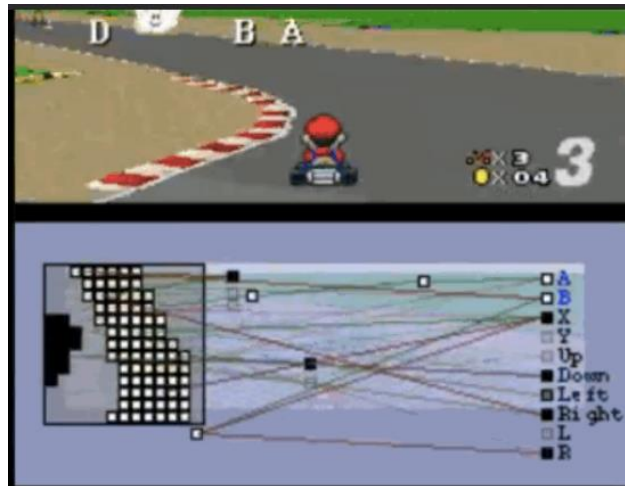
Hoy en día se utiliza para:

- Autos sin conductor
- Aspiradoras robot
- Juegos
- Comercio automatizado
- Gestión de recursos empresariales

Entre los algoritmos más populares: Q-Learning, SARSA, DQN, A3C, Algoritmos Genéticos

Finalmente llegamos a algo que parece inteligencia artificial real. En muchos artículos se coloca aprendizaje por refuerzo en algún punto intermedio entre el aprendizaje supervisado y no supervisado. ¡Ellos no tienen nada en común! ¿Acaso lo dicen por el nombre?

El aprendizaje por refuerzo se utiliza en casos en los que el problema no está relacionado con los datos, pero tiene un entorno en el que vivir: como un mundo de videojuegos o una ciudad para un automóvil autónomo.



El conocimiento de todas las reglas de las calles del mundo no le enseñará a un piloto automático como conducir en ellas. Independientemente de la cantidad de datos que recopilamos, aún no podemos prever todas las situaciones posibles. Es por eso que su objetivo es minimizar el error, no predecir todos los movimientos.

Sobrevivir en un entorno es la idea central del aprendizaje por refuerzo. Lanza al pobre y pequeño robot a la vida real, castígalo por errores y recompénsalo por las acciones correctas. De la misma manera que enseñamos a nuestros hijos, ¿verdad?

Una forma más efectiva sería construir una ciudad virtual y dejar que el automóvil autónomo aprenda todos los trucos allí antes de llevarlo a la ciudad real. Así es exactamente como entrenamos a los pilotos automáticos en este momento. Crea una ciudad virtual basada en un mapa real, puebla con peatones y deja que el auto aprenda a matar a la menor cantidad de personas posible. Cuando el robot tiene una confianza razonable en este mundo artificial, puedes probarlo en las calles reales. ¡Divertido!

En el aprendizaje por refuerzo suelen haber dos enfoques diferentes: **basado en modelo** y **sin modelo**.

El aprendizaje por refuerzo basado en modelo significa que el automóvil necesita memorizar un mapa o sus partes. Es un enfoque bastante anticuado ya que es imposible que el pobre auto sin conductor memorice todo el planeta.

En el aprendizaje sin modelos, el automóvil no memoriza cada movimiento, sino que trata de generalizar situaciones y actuar racionalmente mientras obtiene una recompensa máxima.

HOW MACHINES BEHAVE IN CASE OF FIRE

CLASSICAL PROGRAMMING

«I counted all the scenarios, and now you have to take off your underwear and make a rope of it»

MACHINE LEARNING

«According to my statistics, people die in 6% of fires. So I recommend you to die now»

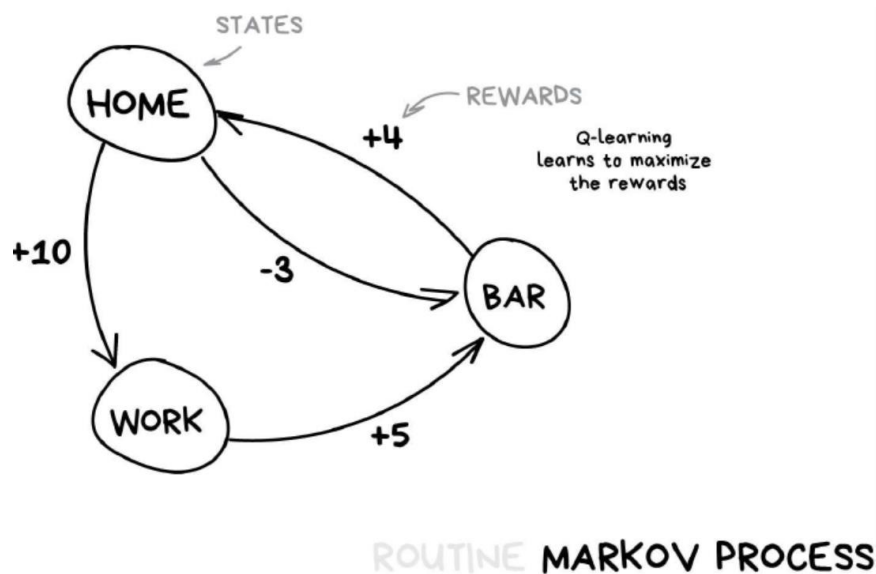
REINFORCEMENT LEARNING

«JUST RUN FOR YOUR FREAKING LIFE AAAAAAAA!!!!»

¿Recuerdas la noticia de que la Inteligencia Artificial venció al mejor jugador Go?. Esto sucedió a pesar de que poco antes se demostró que la cantidad de combinaciones en este juego es mayor que la cantidad de átomos en el universo.

Esto último significa que la máquina no podría ser capaz de recordar todas las combinaciones. Sin embargo, logró ganarle al mejor (como también pasó con el ajedrez). En cada turno, simplemente eligió el mejor movimiento para cada situación y lo hizo tan bien que logró ganarle a una persona.

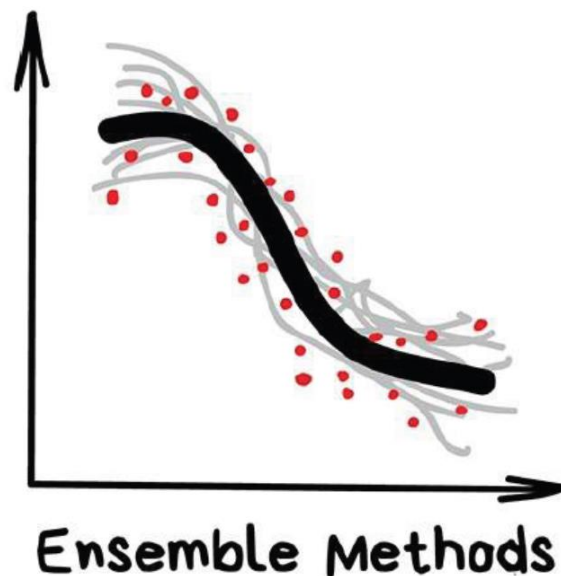
Este enfoque es un concepto central detrás de Q-learning y sus derivados (SARSA y DQN). 'Q' en el nombre significa "Calidad" (en inglés) de la acción en cada situación y en las que todas las situaciones se memorizan como un simple proceso markoviano.



Dicha máquina puede probar miles de millones de situaciones en un entorno virtual, recordando qué soluciones condujeron a una mayor recompensa. Pero, ¿cómo puede distinguir situaciones vistas anteriormente de una completamente nueva?. Si un auto sin conductor se encuentra en un cruce de carreteras y el semáforo se pone verde, ¿significa que puede arrancar ahora? ¿Qué pasa si hay una ambulancia corriendo por una calle cercana?

La respuesta hoy es "nadie lo sabe". No hay una respuesta fácil. Los investigadores lo buscan constantemente pero, mientras tanto, solo encuentran soluciones alternativas. Algunos codificarían manualmente todas las situaciones que les permitieran resolver casos excepcionales. Otros profundizarían y dejarían que las redes neuronales hicieran el trabajo de resolverlo. Esto nos llevó a la evolución del Q-learning llamado Deep Q-Network (DQN). Pero tampoco son la bolita mágica.

Parte 3: Métodos de Conjuntos



"Montón de estúpidos árboles aprendiendo a corregir errores unos de otros"

Hoy en día se utiliza para:

- Todo lo que se ajusta a los enfoques de algoritmos clásicos (pero funciona mejor)
- Sistemas de búsqueda (★)
- Visión por computador
- Detección de objetos

Entre los algoritmos más populares: Random Forest, Gradient Boosting

Es hora de métodos modernos y para "adultos". Los métodos de conjuntos y las redes neuronales son los dos luchadores principales que allanan nuestro camino hacia una singularidad. Hoy están produciendo los resultados más precisos y son ampliamente utilizados en la producción.

Sin embargo, las redes neuronales tienen todo el entusiasmo hoy, mientras que las palabras como "impulsar" o "embolsar" son hipsters escasos en TechCrunch.

A pesar de toda la efectividad, la idea detrás de esto es demasiado simple. Si toma un montón de algoritmos ineficientes y los obliga a corregir los errores de cada uno, la calidad general de un sistema será más alta que incluso los mejores algoritmos individuales.

Obtendrá resultados aún mejores si toma los algoritmos más inestables que predicen resultados completamente diferentes con pequeñas variaciones en los datos de entrada, tal como los árboles decisión y la regresión.

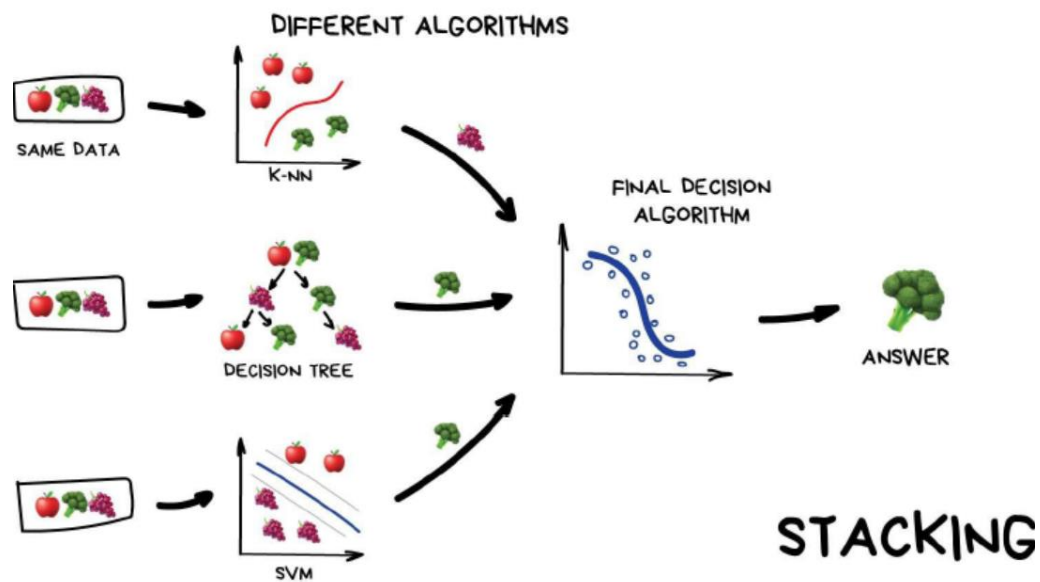
Justo esto es lo que necesitamos.

Podemos usar cualquier algoritmo que conozcamos para crear un Método de Conjuntos. Simplemente coloque un montón de clasificadores, condimente con una regresión y no olvide

medir la precisión. Desde mi experiencia: ni siquiera pruebes un Bayes o kNN aquí que, aunque "tontos", son realmente estables. Eso es aburrido y predecible. Algo así como tu ex.

En su lugar, hay tres métodos probados en grandes batallas para crear Métodos de Conjuntos.

El **Apilamiento** de la salida de varios modelos se pasa en paralelo como entrada a uno último que es capaz de tomar la decisión final. Como esa chica que les pregunta a sus amigas si debe ser tu novia, aunque la decisión final la toma ella misma.



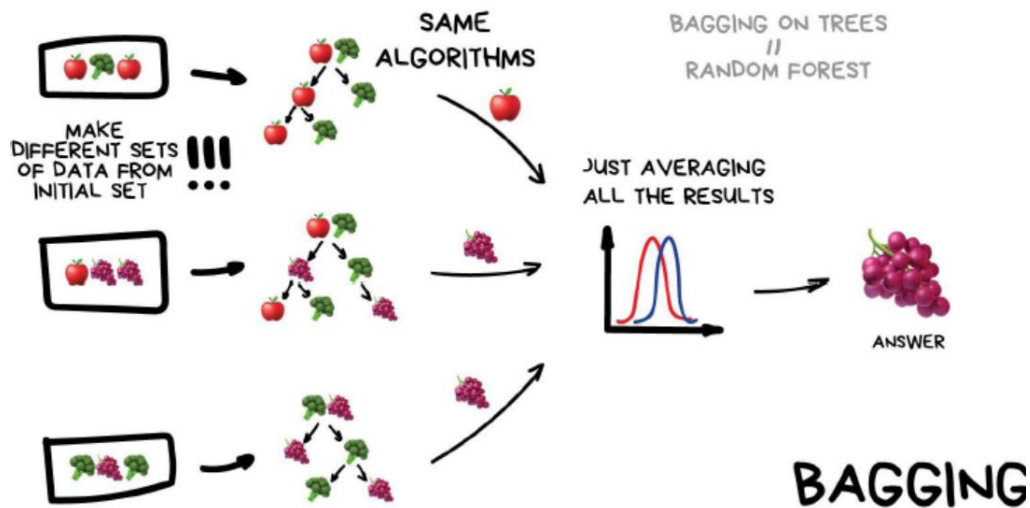
Hay que hacer énfasis aquí en la palabra "diferente". Mezclar los mismos algoritmos con los mismos datos no tendría sentido. La elección de los algoritmos depende completamente de usted. Sin embargo, para el modelo que toma las decisiones finales, la regresión suele ser una buena opción.

Según mi experiencia, el apilamiento es menos popular en la práctica, porque otros dos métodos están dando una mayor precisión.

Bagging aka **Bootstrap AGG**regatING. Use el mismo algoritmo pero entrénelo en diferentes subconjuntos de datos originales.

Al final, ajuste de acuerdo a la combinación de respuestas.

Los datos en subconjuntos aleatorios pueden repetirse. Por ejemplo, de un conjunto como "1-2-3" podemos obtener subconjuntos como "2-2-3", "1-2-2", "3-1-2" y así sucesivamente. Utilizamos estos nuevos conjuntos de datos para enseñar el mismo algoritmo varias veces y luego predecir la respuesta final a través de una votación por mayoría simple.



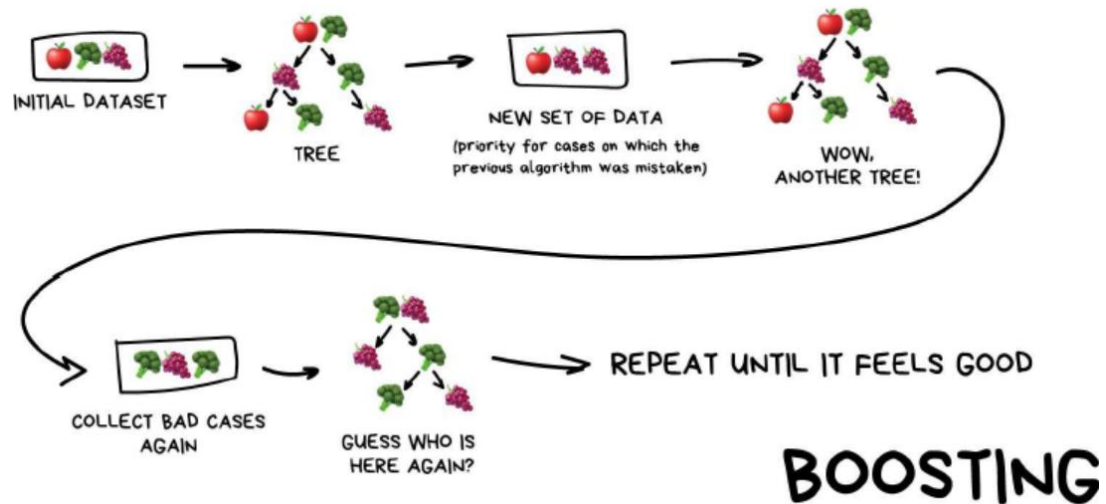
El ejemplo más famoso de Bagging es el algoritmo Random Forest, que es simplemente un Bagging de árboles de decisión (vistos anteriormente). Cuando abres la aplicación de la cámara de tu teléfono y las ves dibujando cuadros alrededor de las caras de las personas, probablemente sea el resultado del trabajo de Random Forest. Las redes neuronales serían demasiado lentas para ejecutarse en tiempo real, pero el bagging es ideal dado que puede calcular árboles en todos los shaders de una tarjeta de video o en estos nuevos y sofisticados procesadores para el aprendizaje automático.



En algunas tareas, la capacidad de ejecutarse en paralelo del Random Forest es más importante que una pequeña pérdida de precisión con respecto al boosting, por ejemplo. Especialmente en procesamiento en tiempo real. Siempre hay una compensación.

Los algoritmos de **Boosting** se entrenan uno por uno secuencialmente. Cada uno de los siguientes prestará la mayor parte de su atención a los puntos de datos que el anterior predijo erróneamente. Repite hasta que estés feliz.

Al igual que bagging, utilizamos subconjuntos de nuestros datos, pero esta vez no se generan aleatoriamente. Ahora, en cada submuestra tomamos una parte de los datos que el algoritmo anterior no pudo procesar. Por lo tanto, hacemos que un nuevo algoritmo aprenda a corregir los errores del anterior.

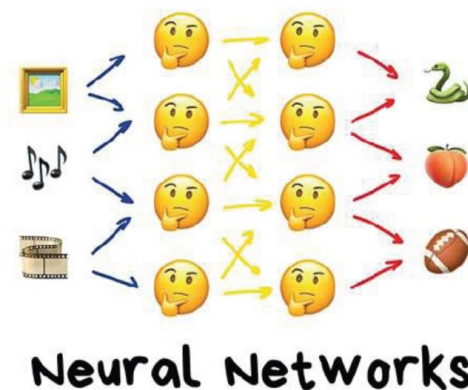


La principal ventaja aquí es que hay una precisión de clasificación muy alta. Los inconvenientes ya se mencionaron, no se pueden paralelizar. Pero sigue siendo más rápido que las redes neuronales. Es como una carrera entre un camión de volteo y un auto de carreras. El camión puede hacer más, pero si quiere ir rápido, tome un automóvil.

Si desea un ejemplo real de boosting, abra Facebook o Google y comience a escribir en el campo de búsqueda. ¿Puedes escuchar un ejército de árboles rugiendo y trabajando juntos para ordenar los resultados por relevancia? Eso es porque están usando boosting.

Hoy en día hay tres herramientas populares para boosting: CatBoost, LightGBM y XGBoost.

Parte 4: Redes Neuronales y Aprendizaje Profundo



"Tenemos una red de mil capas, docenas de tarjetas de video, pero aún no tenemos idea de dónde usarla. ¡Generemos fotos de gatos!"

Usado hoy para:

- Reemplazo de todos los algoritmos anteriores
- Identificación de objetos en fotos y videos
- Reconocimiento y síntesis de voz.
- Procesamiento de imagen, transferencia de estilo
- Traducción automática

Entre las arquitecturas más populares: Perceptron, Redes Convolucionales (CNN), Redes recurrentes (RNN), Autoencoders.

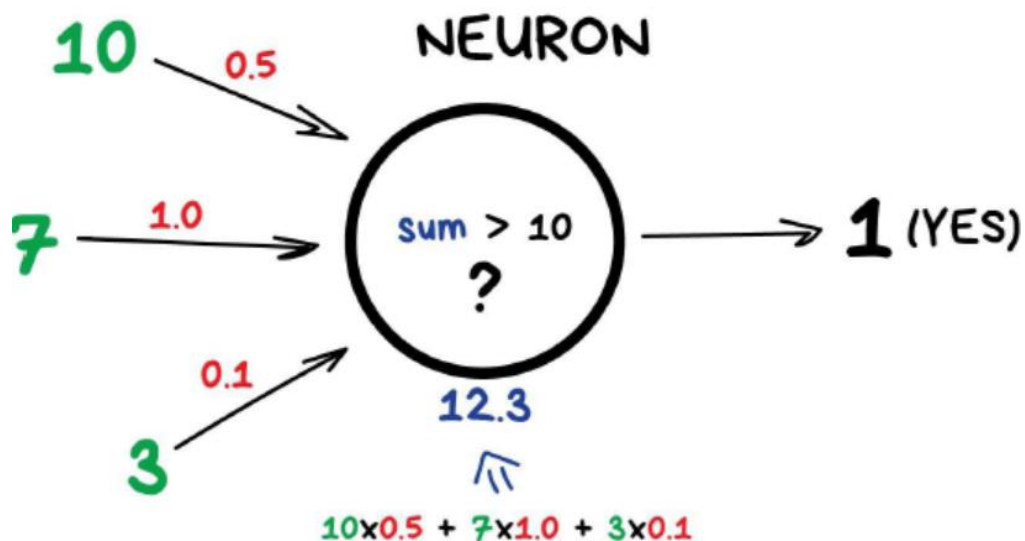
Si alguien le ha tratado de explicar las redes neuronales sin usar las analogías con el "cerebro humano", eres feliz. Cuéntame ese secreto. Pero primero, déjenme explicarlo de la manera que me gusta.

Cualquier red neuronal es básicamente una colección de neuronas y conexiones entre ellas. **Neuron** es una función con un montón de entradas y una salida. Su tarea es tomar todos los números de la entrada, transformarlos a través de una función y enviar el resultado a la salida.

Aquí hay un ejemplo de una neurona simple pero útil en la vida real: suma todos los números de las entradas y, si esa suma es mayor que N, da 1 como resultado. De lo contrario, cero.

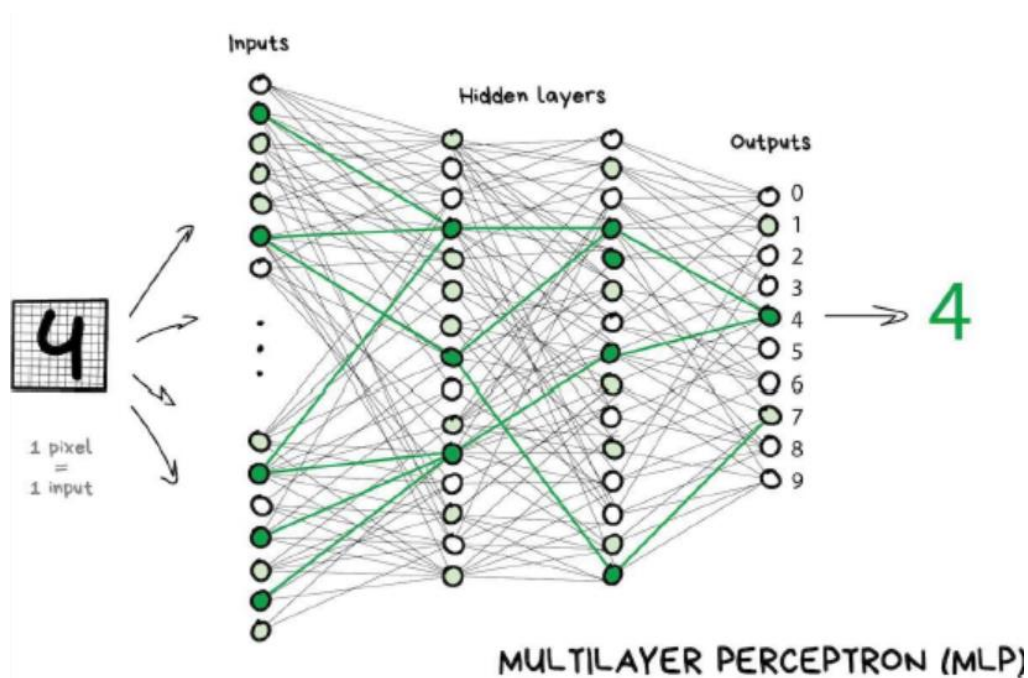
Las **conexiones** son como canales entre neuronas. Conectan las salidas de una neurona con las entradas de otra neurona para que puedan enviarse dígitos entre sí. Cada conexión tiene un solo parámetro llamado **peso**. Es como la fuerza de la señal en la conexión neuronal. Cuando el número 10 pasa a través de una conexión con un peso de 0.5, se convierte en 5.

Estos pesos le dicen a la neurona que responda más a una entrada y menos a otra. Los pesos se ajustan durante el entrenamiento, así es como aprende la red. Básicamente, eso es todo lo que hay que hacer.



Para evitar que la red caiga en la anarquía, las neuronas están unidas por capas, no al azar. Dentro de una misma capa, las neuronas no están conectadas, pero están conectadas a las neuronas de la capa siguiente y de la anterior. Los datos en la red van estrictamente en una dirección: desde las entradas en la primera capa hasta las salidas en la última capa.

Si agrega una cantidad suficiente de capas y coloca los pesos correctamente, obtendrá lo siguiente: aplicando a la entrada, digamos una imagen del número 4 escrito a mano, los píxeles negros de la imagen activarán a las neuronas asociadas que, a su vez, provocan la activación de algunas neuronas de la siguiente capa y así sucesivamente por toda la red hasta que finalmente en la salida se activa la neurona a cargo de reconocer los números cuatro (4). El resultado se logra.



Al programar en la vida real, nadie escribe neuronas y conexiones. En su lugar, todo se representa como matrices y se calcula en función de la multiplicación de matrices para un mejor rendimiento. Mira mi video favorito sobre esto para entender todo el proceso de reconocimiento de dígitos escrito a mano: <https://vimeo.com/137179005>

Una red que tiene múltiples capas se llama Perceptrón Multicapa (MLP) y se considera la arquitectura más simple. Solo para un novato, no la vi usada para resolver tareas en producción.

Después de construir una red, nuestra tarea es lograr formas adecuadas para que las neuronas reaccionen correctamente a las señales entrantes. Ahora es el momento de recordar que tenemos datos que son muestras de 'entradas' y 'salidas' adecuadas. Le mostraremos a nuestra red un dibujo del mismo dígito 4 y le diremos 'adapta tus pesos para que cada vez que veas esta entrada tu salida emita 4'.

Para empezar, todos los pesos se asignan al azar. Después de mostrarle un dígito, emite una respuesta aleatoria porque los pesos aún no son correctos y comparamos cuánto difiere este resultado del correcto. Luego, comenzamos a atravesar la red hacia atrás desde las salidas a las

entradas y le decimos a cada neurona 'hey, te activaste aquí pero hiciste un trabajo terrible y el resultado es inútil. Tu tendrás poca fuerza (peso) en tu conexión para tomar en cuenta otras más importantes, ¿vale? '.

Después de cientos de miles de ciclos de 'inferir-verificar-castigar', existe la esperanza de que los pesos se corrijan para que actúen según lo previsto. El nombre científico para este enfoque es Backpropagation o "método de retropropagación del error".

Una red neuronal bien entrenada puede simular el trabajo de cualquiera de los algoritmos descritos en este artículo (y con frecuencia funciona con mayor precisión). Esta universalidad es lo que las hizo ampliamente populares. Finalmente, tenemos una arquitectura del cerebro humano sobre la que solo necesitábamos agregar muchas capas y enseñarles cualquier tipo de dato que pudiera esperar. A pesar de este gran logro, llegaron las desilusiones. Estábamos entre los 80 y 90.

Resultó que las redes que tenían una gran cantidad de capas requerían un poder de cálculo inimaginable. A pesar de que hoy en día cualquier PC con geforces supera a los centros de datos de esa época, la gente no tenía ninguna esperanza de adquirir una potencia de cálculo como esa y las redes neuronales eran una gran pérdida de tiempo.

Y luego, hace diez años, se propone el término de aprendizaje profundo.

Hay una buena línea de tiempo en el aprendizaje automático que describe la montaña rusa de esperanzas y olas de pesimismo.

En 2012, las redes neuronales convolucionales obtuvieron una victoria abrumadora en la competencia ImageNet que hizo que el mundo repentinamente recordara los métodos de aprendizaje en redes neuronales descritos en los antiguos años 90. ¡Ahora tenemos tarjetas de video!

Las diferencias entre el aprendizaje profundo y las redes neuronales clásicas estaban en los nuevos métodos de entrenamiento que podían manejar redes más grandes. Hoy en día solo la teoría trataría de dividir qué aprendizaje considerar, profundo o no tan profundo. Y nosotros, como profesionales, estamos usando bibliotecas 'profundas' populares como Keras, TensorFlow y PyTorch, incluso cuando construimos una mini red con cinco capas.

Te comentaré sobre dos tipos principales que hay hoy en día.

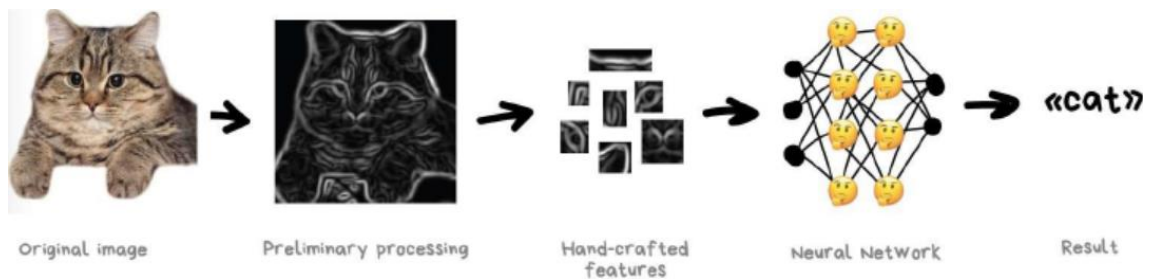
Redes Neuronales Convolucionales (CNN)

Las redes neuronales convolucionales están de moda en este momento. Se utilizan para buscar objetos en imágenes y videos, reconocimiento de rostros, transferencia de estilos, generación y mejora de imágenes, creando efectos de cámara lenta y mejorando la calidad de las imágenes. Hoy en día, las CNN se utilizan en todos los casos que involucran imágenes y videos. Incluso en su iPhone, varias de estas redes atraviesan sus desnudos para detectar objetos en ellas. Si es que hay algo que detectar, jeje.



La imagen de arriba es un resultado producido por **Detectron** que fue recientemente abierto por Facebook,

Un problema con las imágenes siempre fue la dificultad de extraer características de ellas. Puede dividir el texto por oraciones, buscar atributos de palabras en vocabularios especializados, etc. Pero las imágenes tenían que etiquetarse manualmente para enseñarle a la máquina dónde estaban las orejas de gato o la cola en una imagen específica. Este enfoque recibió el nombre de 'artesanía de características' y solía ser utilizado por casi todos.



Hay muchos problemas con la artesanía.

En primer lugar, si un gato tenía las orejas hacia abajo o se alejaba de la cámara: estás en problemas, la red neuronal no verá nada.

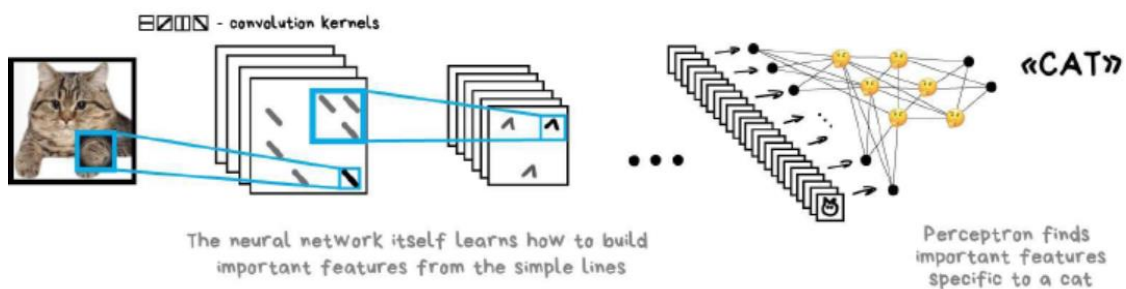
En segundo lugar, intente nombrar en el acto 10 características diferentes que distinguen a los gatos de otros animales. Por mi parte, no podría hacerlo, pero cuando veo una mancha negra corriendo por mi lado por la noche, incluso si solo la veo por el rabillo del ojo, definitivamente diría que es un gato o una rata. Esto se debe que las personas no solo miran la forma de la oreja o el número de patas, sino que cuentan muchas características diferentes en las que ni siquiera piensan. Y así no se lo puede explicar a la máquina.

Esto significa que la máquina necesita aprender tales características por sí misma, basándose en líneas básicas. Haremos lo siguiente: primero, dividimos la imagen completa en bloques de 8x8

píxeles y asignamos a cada uno un tipo de línea dominante, ya sea horizontal [-], vertical [|] o una diagonal [/]. También podemos asignar intensidad de gris de acuerdo a su visibilidad.

La salida sería varias tablas de líneas que, de hecho, son las características más simples que representan los bordes de los objetos en la imagen. Son imágenes en sí mismas pero construidas con líneas. Entonces, una vez más, podemos tomar un bloque de 8x8 y ver cómo se unen. Esto lo hacemos una y otra vez ...

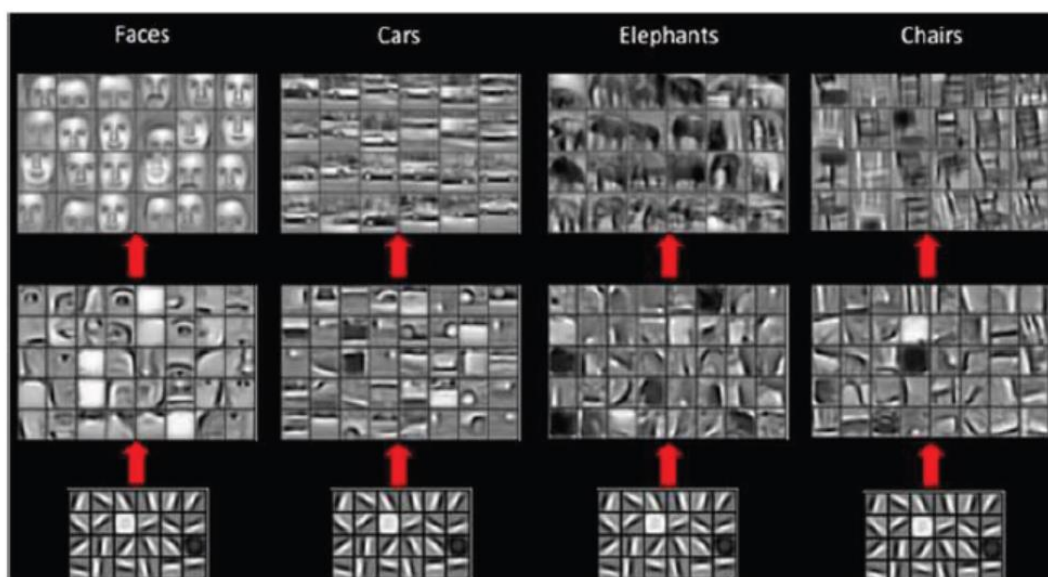
Esta operación se llama convolución, que dio el nombre del método. La convolución puede representarse como una capa de una red neuronal, porque cada neurona puede actuar como una función.



CONVOLUTIONAL NEURAL NETWORK (CNN)

Cuando alimentamos nuestra red neuronal con muchas fotos de gatos, automáticamente asigna pesos mayores a las combinaciones de líneas que vio con mayor frecuencia. No le importa si fue una línea recta de la espalda del gato o un objeto geométricamente complicado como la cara, algo se activará con mucha intensidad.

Como resultado, pondríamos un perceptrón simple que analizará las combinaciones más activadas, el cual se basará en eso para diferenciar a los gatos de los perros, por ejemplo.



La belleza de esta idea es que tenemos una red neuronal que busca las características más distintivas de los objetos por sí misma. No necesitamos elegirlos manualmente. Podemos alimentar cualquier cantidad de imágenes de cualquier objeto simplemente buscando en Google miles de millones de imágenes y nuestra red creará mapas de funciones desde líneas y aprenderá a diferenciar cualquier objeto por sí misma.

Para esto, incluso tengo una broma divertida y divertida:

Dele a su red neuronal un pez y podrá detectar peces por el resto de su vida. Dele a su red neuronal una caña de pescar y podrá detectar cañas de pescar por el resto de su vida ...

Redes Neuronales Recurrentes (RNN)

Es la segunda arquitectura más popular hoy en día. Las redes recurrentes nos dieron cosas útiles como traducción automática, reconocimiento de voz y síntesis de voz en asistentes inteligentes. Los RNN son los mejores para datos secuenciales como voz, texto o música.

¿Recuerdas a Microsoft Sam, el sintetizador de voz de la vieja escuela de Windows XP? Ese tipo divertido construye palabras letra por letra, tratando de unirlos. Ahora, mira Amazon Alexa o Assistant de Google. No solo dicen las palabras con claridad, ¡incluso colocan los acentos correctos!

Los asistentes de voz modernos están entrenados para hablar no letra por letra, sino con frases completas. Podemos tomar un montón de textos sonoros y entrenar una red neuronal para generar una secuencia de audio más cercana al discurso original.

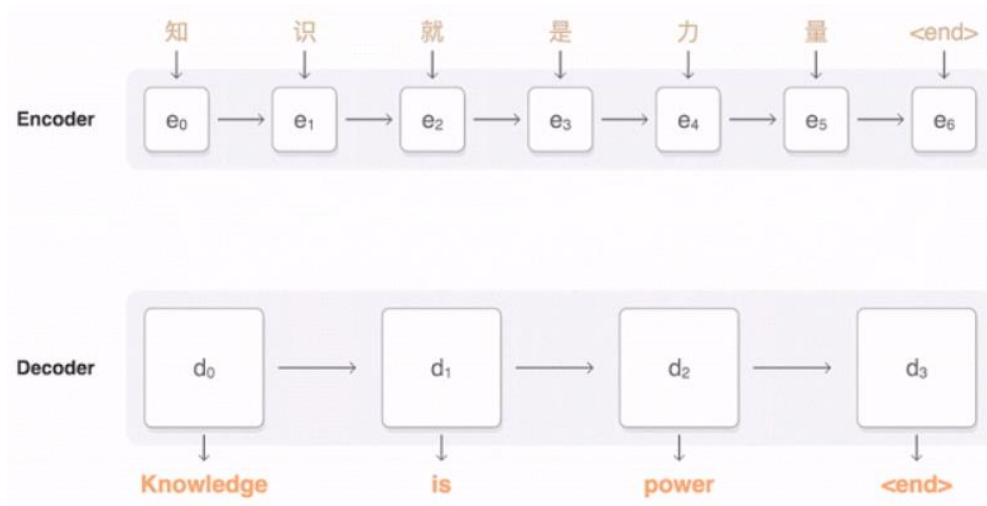
En otras palabras, usamos el texto como entrada y su audio como la salida deseada. Pedimos a una red neuronal que genere algo de audio para el texto dado, luego lo comparamos con los originales, corregimos los errores y tratamos de acercarnos lo más posible al ideal.

Suena como un proceso de aprendizaje clásico. Incluso un perceptrón es adecuado para esto. Pero, ¿cómo deberíamos definir sus resultados? Disparar una salida particular para cada frase posible no es una opción, obviamente.

Aquí nos ayudará el hecho de que el texto, el habla o la música son secuencias. Consisten en unidades consecutivas como sílabas. Todos suenan únicos pero dependen de los anteriores. Pierde esta conexión y obtienes algo tipo dubstep.

Podemos entrenar al perceptrón para generar estos sonidos únicos, pero ¿cómo recordará las respuestas anteriores? Entonces, la idea es agregar memoria a cada neurona y usarla como entrada adicional en la próxima ejecución. Una neurona podría tomar nota por sí misma: hey, teníamos una vocal aquí, el siguiente sonido debería sonar más alto (es un ejemplo muy simplificado).

Así aparecieron las redes recurrentes.

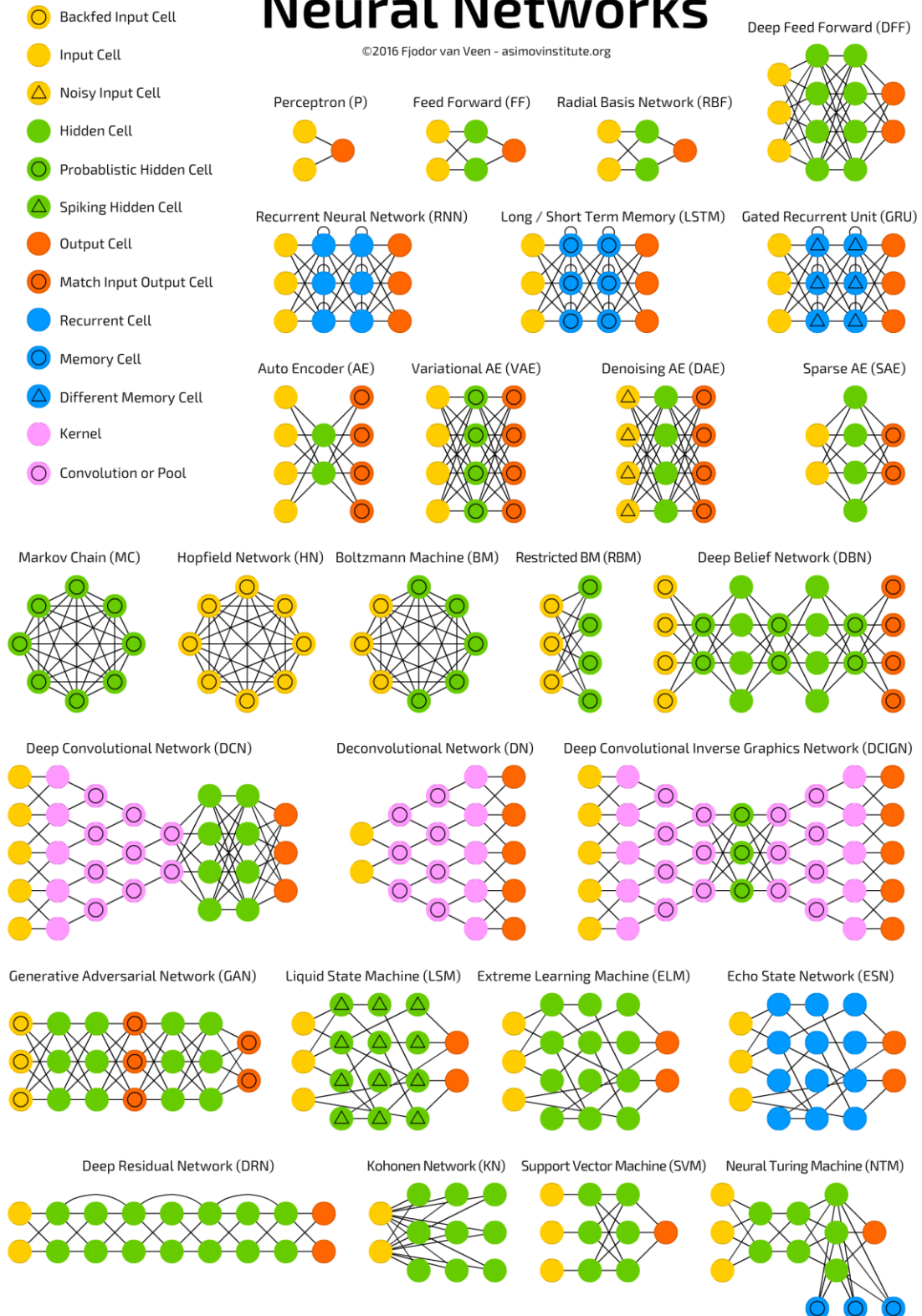


Hay muchas más arquitecturas de red en la naturaleza. Recomiendo un buen artículo llamado **Neural Network Zoo**, donde se recopilan y explican brevemente casi todos los tipos de redes neuronales.

A mostly complete chart of

Neural Networks

©2016 Fjodor van Veen - asimovinstitute.org



El fin: ¿Cuándo es la guerra con las máquinas?

El principal problema aquí es que la pregunta "¿cuándo serán las máquinas más inteligentes que nosotros y nos esclavizarán a todos?" Inicialmente está mal. Hay demasiadas condiciones ocultas en ella.

Decimos "conviértete más inteligente que nosotros" como si quisiéramos decir que hay una cierta escala unificada de inteligencia en la que en la parte superior de la inteligencia está el ser humano, los perros un poco más abajo y las estúpidas palomas merodean por la parte inferior.

Eso está mal.

Si este fuera el caso, cada humano debe vencer a los animales en todo, pero no es cierto. La ardilla promedio puede recordar mil lugares escondidos con nueces; y yo ni siquiera puedo recordar dónde están mis llaves.

Entonces, ¿la inteligencia es un conjunto de habilidades diferentes, no un solo valor medible? ¿O recordar lugares escondidos de nueces no está incluido en la inteligencia?

Una pregunta aún más interesante para mí: ¿por qué creemos que las posibilidades del cerebro humano son limitadas?

Hay muchos gráficos populares en Internet donde el progreso tecnológico se dibujan de manera exponencial y las posibilidades humanas como constantes. ¿Pero es así?

Bien, multiplique 1680 por 950 ahora mismo en su mente. Sé que ni siquiera lo intentará, perezosos bastardos. Pero déle una calculadora: lo hará en dos segundos. ¿Significa esto que la calculadora solo expandió las capacidades de su cerebro?

En caso afirmativo, ¿puedo continuar con ejemplos de otras máquinas? Como, ¿usar notas en mi teléfono para no recordar ciertos datos? Oh, justo es lo que hago ahora. Estoy ampliando las capacidades de mi cerebro con las máquinas.

Piénsalo. Gracias por leer.