

Programación Web 3

UNLaM - Tecnicatura en Desarrollo Web

Trabajo Práctico de Investigación

Inteligencia Artificial en .NET

| | |
|--|----------|
| Integrantes..... | 2 |
| Objetivo..... | 2 |
| Objetivos de la cátedra:..... | 2 |
| Objetivos del grupo:..... | 2 |
| Situación Actual..... | 3 |
| Desarrollo de la Investigación..... | 3 |
| Memoria descriptiva..... | 4 |
| Conclusiones..... | 6 |
| Referencias/Bibliografía..... | 6 |

Integrantes

- Giménez Agustina
- Fernández Santiago
- Chilon Eduar
- Ramos Juan Manuel
- Castillo Emiliano

Objetivo

Objetivos de la cátedra:

Entender y diferenciar los conceptos IA: Inteligencia Artificial, Machine Learning, Deep Learning. Encontrar frameworks o librerías .Net que permitan Deep Learning: Reconocimiento de voz, reconocimiento de imágenes, etc. Ejemplo de Librerías: Accord, Aforge.Net, ML.Net, etc.

Objetivos del grupo:

- Familiarizarnos con conceptos y el funcionamiento acerca de la inteligencia artificial.
- Desarrollar una aplicación .NET implementando inteligencia artificial a través de ML.NET y que por la cual se pueda realizar reconocimiento de imágenes, persistir sus datos y utilizarlos a lo largo del proyecto.

Situación Actual

¿Qué es la inteligencia artificial?

Entendemos como Inteligencia Artificial a una disciplina dentro de las ciencias de la computación que consiste en el desarrollo de un tipo de algoritmos en el ámbito del software y del hardware para ejecutarlos de la manera más eficiente posible tanto con respecto al consumo energético como de tiempo de ejecución.

La mayoría de algoritmos de IA existentes hacen el uso del hardware para resolver ciertos tipos de problemas, pero lo hacen de una manera concreta. A través de unos datos de entrada, el algoritmo aprende a clasificar la información o a hacer predicciones a través de patrones concretos. Es algo que da la apariencia de ser mágico, pero lo que hace es utilizar esos datos para afinar su predicción, siendo esta predicción el algoritmo, el cual en vez de ser estático va variando a medida que entran nuevos datos y se acaba viendo si la predicción es correcta o no.

Tipos de inteligencia artificial

Lo que diferencia a la Inteligencia Artificial de otros programas de ordenador es que no hay que programarla específicamente para cada escenario. Podemos enseñarle cosas (Machine Learning, aprendizaje automático), pero también puede aprender por sí mismo (Deep Learning).

Machine Learning

En la década de los años 80 apareció Machine Learning (ML), que es una rama de la Inteligencia Artificial, que consiste en la capacidad de las máquinas de aprender por sí mismas basándose en datos e identificación de conductas, lo que les permite re-entrenarse y cualificar las decisiones de forma autónoma, es decir, sin la necesidad de la supervisión humana.

Intentando simplificar mucho su funcionamiento, básicamente un algoritmo analiza un gran volumen de datos, detecta patrones de comportamiento y saca conclusiones permitiendo eficientar el proceso de manera automática.

Un ejemplo de Machine Learning (ML) en el mundo del marketing, es la personalización de sugerencias en base a tu histórico, como hace Netflix, que te recomienda series o películas basándose en tus gustos o Amazon que te posiciona productos en su web o app en base a tu historial de compras y navegación.

Deep Learning

El Deep Learning (DL) surgió en el siglo XXI, y se trata de algoritmos basados en redes neuronales artificiales que imitan el funcionamiento del cerebro humano, siendo la técnica que más se asemeja al aprendizaje de las personas.

El Deep Learning (DL) permite solucionar problemas complejos, aunque el conjunto de datos sea no estructurado y no exista una relación clara entre ellos, cuanto más se ejecuten y aprendan los algoritmos de Deep Learning, mejor es el resultado obtenido. Este aspecto es un valor diferencial respecto del Machine Learning, que requiere que los datos hayan sido previamente estructurados para ofrecer su máximo rendimiento.

Algunos ejemplos de aplicaciones prácticas de Deep Learning son las traducciones automáticas donde los algoritmos usan grandes volúmenes de datos para entrenar el modelo permitiendo la interpretación de idiomas o por ejemplo, el reconocimiento de imágenes utilizado en el etiquetado de fotos en redes sociales.--

ML.NET

ML.NET es un framework de aprendizaje automático desarrollado por Microsoft que permite a los desarrolladores de software integrar fácilmente capacidades de aprendizaje automático en sus aplicaciones. Proporciona una forma accesible y eficiente de construir modelos de aprendizaje automático utilizando el lenguaje de programación C#.

ML.NET ha ganado popularidad y aceptación en la comunidad de desarrolladores debido a su enfoque en la simplicidad y la integración con el ecosistema de .NET. Ofrece una amplia gama de algoritmos de aprendizaje automático, como clasificación, regresión, agrupación y detección de anomalías,

entre otros. También proporciona herramientas para la preparación de datos, evaluación de modelos y personalización de los flujos de trabajo de aprendizaje automático.

En términos de funcionalidad, ML.NET ofrece soporte para la implementación de modelos entrenados tanto localmente como en la nube, lo que brinda flexibilidad a los desarrolladores para adaptarse a sus necesidades. Además, se integra bien con otras bibliotecas y herramientas populares de aprendizaje automático, como TensorFlow y ONNX, lo que permite a los desarrolladores aprovechar modelos pre-entrenados y obtener mejores resultados.

Es importante destacar que ML.NET se ha mantenido activo y se han realizado varias actualizaciones y mejoras desde su lanzamiento inicial. Microsoft ha estado trabajando continuamente en el desarrollo y la expansión de ML.NET, agregando nuevas características y funcionalidades, así como corrigiendo problemas y proporcionando documentación y ejemplos útiles para los desarrolladores.

Desarrollo de la Investigación

La clasificación de imágenes es una aplicación popular del aprendizaje automático en la que el objetivo es categorizar imágenes en diferentes clases o etiquetas. ML.NET proporciona funcionalidades y APIs incorporadas para tareas de clasificación de imágenes. Admite diversas arquitecturas de aprendizaje profundo, incluyendo modelos pre-entrenados como ResNet, Inception, y más. ML.NET facilita el entrenamiento de un modelo de clasificación de imágenes utilizando tu propio conjunto de datos personalizado o incluso el aprendizaje por transferencia ajustando modelos pre-entrenados.

ML.NET proporciona una herramienta de interfaz gráfica llamada ModelBuilder para simplificar el proceso de construcción de modelos de aprendizaje automático. Está diseñado para desarrolladores con poca o ninguna experiencia en aprendizaje automático, lo que facilita la creación de modelos para escenarios comunes. Con Model Builder, puedes experimentar rápidamente con diferentes algoritmos, transformaciones de datos y arquitecturas de modelos para encontrar el mejor enfoque para tu problema.

Pasos para preparar y entrenar el modelo:

1. Preparar el conjunto de datos: Recopilar y organizar un conjunto de datos de imágenes etiquetadas. Cada imagen debe estar asociada a una clase o etiqueta específica.
2. Cargar y preprocesar el conjunto de datos: Utilizar las APIs de carga y transformación de datos de ML.NET para cargar el conjunto de datos en la memoria y aplicar los pasos de preprocesamiento necesarios, como el redimensionamiento, la normalización o el aumento de datos.
3. Definir el flujo de trabajo del modelo: Utilizar Model Builder o las APIs de programación de ML.NET para definir la estructura del modelo de clasificación de imágenes. Esto incluye especificar el tamaño de imagen de entrada, seleccionar la arquitectura o el modelo pre-entrenado, y configurar cualquier capa adicional o transformación.
4. Entrenar el modelo: Utilizar el conjunto de datos preparado para entrenar el modelo de clasificación de imágenes. ML.NET proporciona APIs para realizar el entrenamiento, donde se especifican los datos de entrada, el flujo de trabajo del modelo, los parámetros de entrenamiento (por ejemplo, número de iteraciones, tasa de aprendizaje) y las métricas de evaluación.
5. Evaluar el modelo: Después del entrenamiento, evaluar el rendimiento del modelo utilizando métricas de evaluación como la precisión, la exhaustividad, la recuperación o la puntuación F1. Esto ayuda a evaluar qué tan bien el modelo generaliza en nuevas imágenes no vistas.
6. Utilizar el modelo para hacer predicciones: Una vez que el modelo está entrenado y evaluado, se puede utilizar para hacer predicciones en nuevas imágenes. ML.NET proporciona APIs para cargar el modelo entrenado y realizar predicciones en una o varias imágenes.

Memoria descriptiva

Para este proyecto, comenzamos generando una clasificación de posibles emociones (feliz, enojado, triste), en diferentes carpetas, es decir, recopilamos un conjunto de imágenes relacionadas con cada emoción y las organizamos en carpetas separadas. Una vez que las carpetas tuvieron una gran cantidad de imágenes se procedió a entrenar la inteligencia. Así, ML NET puede detectar a través de patrones que existen en esas carpetas, cómo deberá clasificar imágenes que obtenga en un futuro. Una vez que entrenamos el modelo, automáticamente se genera un código para ser utilizado como api y así poder consumirlo desde nuestro proyecto.

Una vez que pudimos clasificar las emociones según imágenes que subimos, procedimos a incorporar la cámara web (mediante javascript) en una nueva vista. Una vez que se enciende la cámara, con un botón “Capturar” se toma una foto, la cual es clasificada por la inteligencia de igual manera que si fuera una imagen importada.

Todas las imágenes y sus respectivas clasificaciones (feliz, enojado, etc) son persistidas en una base de datos SQL server.

Por último, creamos una vista de Resultados donde se obtendrá los datos persistidos en la base de datos.

Conclusiones

Durante el desarrollo de este proyecto, nos propusimos familiarizarnos con los conceptos de Machine Learning y reconocimiento de imágenes utilizando la librería ML.Net. Nuestro objetivo principal fue desarrollar una aplicación que pudiera clasificar imágenes en base a emociones específicas, como felicidad, enojo y tristeza.

Aprendimos a utilizar ML.NET para entrenar un modelo, y pudimos aplicarlos con éxito en la clasificación de imágenes según diferentes emociones. Este proyecto nos brindó una comprensión más profunda de los conceptos de Machine Learning y su aplicación en el reconocimiento de imágenes, así como la capacidad de utilizar ML.Net de manera efectiva en futuros proyectos relacionados con este ámbito.

Referencias/Bibliografía

<https://dotnet.microsoft.com/en-us/apps/machinelearning-ai/ml-dotnet>

<https://github.com/dotnet/machinelearning>

<https://docs.microsoft.com/es-es/dotnet/machine-learning/>

<https://hipertextual.com/2023/02/diferencias-ia-machine-learning>

<https://hardzone.es/tutoriales/rendimiento/diferencias-ia-deep-machine-learning/>

<https://www.salesforce.com/mx/blog/2017/6/Que-es-la-inteligencia-artificial.html>