Guia de usuario

October 6, 2016

1 Introducción

Esta guía de usuario tiene como objetivo entregar los conociemientos necesarios para usar la librería NEAT aquí creada.

Se comenzará explicando la instalación de la librería y el proceso de compilación en proyectos externos, luego se explicará qué es lo mínimo que se debe saber para poder usar esta librería y aprovechar al máximo sus capacidades, luego de estos tópicos los siguientes serán dirijidos a interesados en detalles para lo cual se explicará en resumen qué es NEAT, cuál es el módelo aplicado en esta librería para crear la librería NEAT y todas las consideraciones que deben tener para ajustar manualmente los parametros de usuario que sólo deben ser ajustados por personas que entiendan sus implicaciones.

2 Instalación

Importante: Por el momento la instalación sólo es posible en arquitecturas linux (posiblemente también en MAC), pronto será para toda plataforma a través de CMake.

El modelo adoptado para la compilación e instalación de esta librería es a través del compilador gcc usando Makefiles, lo cual corresponde a el método nativo en que se programa c++ en linux. Para instalar la librería en su computadora debe seguir los siguientes pasos:

1. Entrar a su terminal y posicionarla en la carpeta raiz de este proyecto, en donde usted puede encontrar el Readme.md.

2. dentro de la terminar ejecutar el comando

   ```
   $ make
   ```

   A través de este comando usted compilará el proyecto y obtendrá todos los objetos (archivos con extensión .o) que contienen todas las implementaciones de todas las clases usadas para implementar la librería.

3. finalmente ejecuta el comando

   ```
   $ sudo make install
   ```

   Este último comando requiere de los permisos de super usuario. En este paso lo que se realizó es que se crea la librería no dinámica .so y es movida a una ruta en la cual puede ser encontrada por cualquier proyecto personal, además se hizo lo mismo con los headers para que sean encontrables en el PATH del usuario. Con esto ahora usted puede utilizar esta librería en su propio proyecto a través de incluir en su header lo siguiente.

   ```
   #include <NEATSpikes>
   ```

Con los pasos anteriores ya se tiene instalada la librería, en la siguiente sección se explicará como compilar su propio proyecto.

3 Compilación

En linux existen diferentes tipos de librerías, por un lado están las librerías estandar como iostream, cstdlib, entre otras las cuales usted puede agregar en su proyecto y compilar sin agregar nada en su compilación, en cambio para usar NEAT usted debera agregar el flag -lneatspikes al momento de su compilación.

Por ejemplo, si usted crease un codigo trivial en archivo.cpp el cual no realiza nada pero llama a la librería, como el siguiente.

```
#include <NEATSpikes>
int main()
{
        return 0;
}
```

Este código debe ser compilado:

```
$ g++ archivo.cpp -o ejecutable
```

Esta compilación no se realizará correctamente dado que no se encontrarán las implementaciones de ninguna clase de la librería NEATSpikes, ¿cómo es eso posible?. Para ser claros el flag -lneatspikes es la forma en que el compilador puede encontrar los objetos de la librería dado que esta no es una librería estandar de linux.

Además a lo anterior el estandar usado en el proyecto es c++14 (mucho más avanzado que c++ 98) y que es retrocompatible. Tomando en cuenta esto la correcta forma de compilar sería entonces como la siguiente.

```
$ g++ -std=c++14 archivo.cpp -o ejecutable -lneatspikes
```

Consideraciones: No es razonable compilar a mano sus proyectos, y en general se recomienda el uso de Makefiles o CMake, por lo mismo se le propone que examine los makefiles de las experimentos de muestra que se encuentran en experimentSamples.

Notar: Este programa puede estar en cualquier parte de su computadora y funcionará sin problemas.

Ahora que sabemos como instalar la librería y compilar nuestros propios proyectos, necesitamos saber como crear nuestros propios proyectos, lo cual se explicará en la siguiente sección.

4 Uso básico de NEAT

En esta sección se explicará como usar NEAT para el propósito que usted busque.

Si usted esta leyendo esto es porque desea realizar su propio proyecto, esto puede ser muy variado desde tareas matematicas y estadisticas como aproximación por curvas de datos a cosas que escapan de un análisis sencillo como la generación de caminatas de robots, procesamiento de imagenes, entre otro tipo de aplicaciones son posibles.

Vamos paso a paso desmenusando las cosas que usted debe saber para poder programar su propio experimento.

1. Primero que todo visualicemos nuestro objetivo, veamos las redes neuronales como cajas negras las cuales reciben entradas y dan las salidas que usted desea, por ejemplo imagine que usted busca que una red neuronal que recibe todas las mediciones ambientales de una región y debe determinar la probabilidad que exista o no lluvia el próximo día. Notar que existe una clara determinación de entradas a la red neronal y sus salidas (en este particular caso sólo una salida)

2. Segundo entendamos que estas redes neuronales van a ir mejorando en el tiempo y debemos calificarlas de tal manera que aprendan a mejorar sus calificaciones, por ejemplo imaginemos un problema diferente como la clasificación de manzanas o peras en una linea de transmisión y la entrada de la red neuronal es la imagen completa de una fotografía y la red neuronal debe determinar si lo que aparece en la fotografía es manzana o pera. Se podría calificar la red neuronal según cuantos aciertos realizó cada 100 muestras. En muchos casos es más factible calcular el error cometido por la red neuronal, en tales casos se debe ingenear cómo obtener una calificación a través del error, por ejemplo calificación = error maximo posible - error cometido.

Los dos pasos anteriores son lo único necesario, aunque aún estamos hablando en forma básica sin poner manos en código, todo el código que usted debe desarrollar tiene relación con los dos pasos anteriores.

4.1 Comencemos con el código

Ahora sí empezaremos a hablar en términos técnicos y con códigos c++, y para simplificar todo será siempre pensando en uno de los ejemplos de

muestra que hay, el XOR.

Nuestro objetivo es crear una red neuronal que realice las operaciones del operador lógico XOR.

0 XOR 0 –> 0
1 XOR 0 –> 1
0 XOR 1 –> 1
1 XOR 1 –> 0

Dado que de momento no hemos llegado a entender en detalle, usted debe empezar a través del proyecto template que se encuentra en la carpeta template, realice una copia de la carpeta en cualquier parte su computadora.

Examinando el proyecto Template usted encontrará varios archivos sin extensión, todos ellos usted debe dejar sin modificar en cualquier experimento que usted desee menos el archivo ANNUD, el resto sólo deben ser manipulados si lee completamente este documento y no parcialmente:

1. ANNUD: especificaciones de usuario para la clase ANN (artificial neural network).
2. BNUD: especificaciones de usuario para la clase Basic Neuron.
3. BNSWUD: especificaciones de usuario para la clase Basic Synaptic Weight.
4. LifeUD: especificaciones de usuario para la clase Life .
5. OUD: especificaciones de usuario para la clase Organism .
6. SPECIESUD: especificaciones de usuario para la clase Species .
7. RACEUD: especificaciones de usuario para la clase Race.

En el archivo ANNUD usted debe especificar la cantidad de entradas de la red neuronal y la cantidad de salidas de tal manera que coincida con el experimento a realizar. En este caso XOR recibe dos entradas lógicas y calcula una salida, por lo tanto dentro de ANNUD buscar las lineas inputsAmount y outputsAmount y darle valores 2 y 1 respectivamente (de hecho ya debería estar correcto en el Template).

Además de los archivos sin extensión usted encontrará test.cpp y Makefile.

4.1.1 test.cpp

El código template se compone de tres funciones, main, sendAllOrganismToExperiment y experiment, las cuales se explicarán

1. main: metodo principal el cual va administrando las redes neuronales.
2. sendAllOrganismToExperiment: este metodo se encarga de ir entregando las redes neuronales (en NEAT son tratadas como organismos) al experimento.
3. experiment: Esta es la única parte que debes tocar, es cómo evaluar el organismo.

Los organismos son evaluados respecto a su desempeño en el experimento, en este caso el experimento es qué tan bien XOR son, para ello calcularemos su calificación (en adelante llamaremos **fitness** a la calificación)

5 Resumen de NEAT

Partiremos explicando lo esencial de NEAT,