Bartosz Budnik XX.06.2016 r.

Piotr Komar

Sebastian Grzesiak

Michał Szymański

Mateusz Walczak

***Zastosowania informatyki w gospodarce  
Projekt***

***Deeplearning – ner (nazwy własne)***

Prowadzący:

Dr inż. Tomasz Walkowiak

Politechnika Wrocławska

Spis treści

[1. Wstęp 3](#_Toc449988132)

[2. Instalacja środowiska 3](#_Toc449988133)

[2.1. Wykorzystane środowisko 3](#_Toc449988134)

[2.2. Instalacja 3](#_Toc449988135)

[2.3. Inne wersje 5](#_Toc449988136)

[2.4. Rozwiązywanie problemów 6](#_Toc449988137)

[3. Rezultat prac nad projektem 7](#_Toc449988138)

[4. Instrukcja obsługi 7](#_Toc449988139)

[5. Podsumowanie 8](#_Toc449988140)

[6. Spis ilustracji, listingu i tabel 8](#_Toc449988141)

[7. Literatura 8](#_Toc449988142)

# Wstęp

Celem tego projektu było napisanie programu, który będzie w stanie zidentyfikować nazwy własne w tekście oraz, w wersji bardziej zaawansowanej, dodatkowe informacje o wyrazach czy związkach frazeologicznych. Program miał nauczyć się identyfikować wyrazy z dostarczonej bazy i przekazywać informację użytkownikowi, w jakim stopniu nauczył się poprawnie ustalać nazwy własne. Dodatkowo po nauczeniu programu miała istnieć możliwość podawania dowolnych nazw w postaci parametru. W związku z dużą ilością danych, projekt oparty był o GPU, aby przyspieszyć działanie.

W celu napisania programu, należało wykorzystać rozwiązania z dziedziny deeplearning’u. Jest to gałąź uczenia maszynowego oparta o struktury i algorytmy, które modelują dane na wysokim poziomie abstrakcji przez wiele warstw procesujących, o złożonej strukturze składającej się z wielu nieliniowych transformacji[1].

# Instalacja środowiska

## Wykorzystane środowisko

Projekt został napisany w języku programowania python w wersji 2.7. Program działa w oparciu o framework Keras, korzystający z silnika Tensorflow.

TensorFlow to rozwiązanie od Google typu open source przeznaczone do obliczeń numerycznych w oparciu o CPU lub GPU[2]. Framework ten powstał z myślą o machine learning’u, celem przygotowania sieci neuronowych do wykrywania i rozszyfrowywania wzorców oraz korelacji. Keras z kolei to framework z bibliotekami modularnych sieci neuronowych przeznaczony do projektów machine i deep- learning’owych, umożliwiający korzystanie z silnika innych framework’ów, takich jak Theano czy TensorFlow[3].

Działanie programu zostało sprawdzone na systemie Linux, wersjach Fedora oraz Ubuntu. Do instalacji konieczne będzie połączenie z siecią oraz karta graficzna obsługująca CUDA. Wymagane również będzie konto na stronie Nvidia Developer w celu pobrania dodatkowych bibliotek cudNN[4].

## Instalacja

Instrukcja instalacji zakłada, że w systemie Linux nie ma poprzednich wersji TensorFlow, CUDA lub cudNN. Instalacja została wykonana na „czystym” systemie, więc obejmuje również podstawowe kroki. Wszystkie etapy przedstawione w tej instrukcji, zostały wykonane w takiej samej kolejności jak w środowisku testowym. Poniższe komendy zostały wykonane na systemie Linux 64-bit v14.04. Należy pamiętać, że w przypadku różnicy w wersji oprogramowania pobranego z Internetu, trzeba zmodyfikować odpowiednio niektóre komendy.

Instalację środowiska rozpoczęto od aktualizacji sterowników karty graficznej.

sudo apt-get install nvidia-340

Listing 1 Komenda instalująca sterownik graficzny.

W tym wypadku została zainstalowana wersja 340, jednak można zainstalować inną wersję obsługującą CUDA 7.5. Listę aktualnych wersji sterownika karty graficznej można znaleźć na stronie internetowej <http://www.nvidia.com/object/unix.html> (Stan na dzień 02.05.2016 r.).

Kolejnym krokiem była instalacja PIP.

sudo apt-get install python-pip python-dev

Listing 2 Komenda instalująca PIP.

Następnie instalacja TensorFlow z obsługą GPU; wymaga: Python 2.7, CUDA toolkit 7.5, CuDNN v5. Instalacja ostatnich dwóch elementów od Nvidii zostanie przedstawiona w dalszej części.

sudo pip install --upgrade https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow-0.8.0-cp27-none-linux\_x86\_64.whl

Listing 3 Komenda instalująca TensorFlow.

Są również dostępne inne wersje instalacyjne, jednak zostaną one przedstawione w dalszej części, jako opcjonalne zmiany. W projekcie została wykorzystana komenda umieszczona powyżej.

Po tym kroku zainstalowano CUDA 7.5 oraz cudNN 5.0. Aby pobrać, należy się zarejestrować na stronie <https://developer.nvidia.com/> (Stan na dzień 02.05.2016 r.). Najpierw zainstalowano CUDA 7.5 pobrane ze strony <https://developer.nvidia.com/cuda-downloads> (Stan na dzień 02.05.2016 r.). Należy wybrać kolejno:

Linux → x86\_64 → Ubuntu → 14.04 → deb (local)

Pobrany został plik **cuda-repo-ubuntu1404-7-5-local\_7.5-18\_amd64.deb** ważący w przybliżeniu 1.9 GB. Instalacja CUDA przebiegła w następujący sposób.

sudo dpkg -i cuda-repo-ubuntu1404-7-5-local\_7.5-18\_amd64.deb

sudo apt-get update

sudo apt-get install cuda

Listing 4 Komendy potrzebne do instalacji CUDA.

Na stronie internetowej, z której pobrano ten plik znajduje się również kopia tych komend wzbogacona o obszerniejszy opis w pliku .pdf Installation Guide for Linux po adresem <http://developer.download.nvidia.com/compute/cuda/7.5/Prod/docs/sidebar/CUDA_Installation_Guide_Linux.pdf> (Stan na dzień 02.05.2016 r.), wraz z informacjami o rozwiązaniach możliwych do wystąpienia problemów. W przypadku tej instalacji ten pakiet nie sprawił problemów.

Kolejnym krokiem była instalacja cudNN, które jest dostępne pod adresem <https://developer.nvidia.com/cudnn> (Stan na 02.05.2016 r.). Po kliknięciu przycisku *Download*, wyświetlona została krótka ankieta, na którą można było odpowiedzieć. Następnie po kliknięciu *Proceed to download* i zaznaczniu opcji „*I Agree To the Terms…”* wyświetlona została lista z wyborem odpowiedniej wersji. Wybrano *cuDNN v5 Library for Linux*. Pobrany został plik **cudnn-7.5-linux-x64-v5.0-rc** ważący w przybliżeniu 75 MB, który wypakowano i skopiowano, jak wskazują komendy.

tar xvzf cudnn-7.5-linux-x64-v5.0-rc.tgz

sudo cp cudnn-7.5-linux-x64-v5.0-rc/cudnn.h /usr/local/cuda/include

sudo cp cudnn-7.5-linux-x64-v5.0-rc/libcudnn\* /usr/local/cuda/lib64

sudo chmod a+r /usr/local/cuda/lib64/libcudnn\*

Listing 5 Komendy potrzebne do instalacji cudNN.

Powyższe komendy zakładają, że CUDA została zainstalowana w **/usr/local/cuda**. W przypadku instalacji w innej lokalizacji należy odpowiednio zmodyfikować.

Ostatnim krokiem było ustawienie **LD\_LIBRARY\_PATH** i **CUDA\_HOME** do ***~/.bashrc***. Wykonano ten krok korzystając z komend.

export LD\_LIBRARY\_PATH="$LD\_LIBRARY\_PATH:/usr/local/cuda/lib64"

export CUDA\_HOME=/usr/local/cuda

Listing 6 Komenda ustawiająca zmienne środowiskowe.

W tym kroku również należy zmodyfikować komendy, jeśli lokalizacja CUDA jest inna, aniżeli ta założona w instalacji.

Ten krok kończy instalację. Potem sprawdzono czy TensorFlow został prawidłowo zainstalowany poprzez wykonanie następującego kroku.

python

>>> import tensorflow as tf

Listing 7 Weryfikacja poprawności instalacji TensorFlow poprzez import bibliotek.

Ten etap zweryfikował poprawną instalację, więc można było wykonać kolejny krok jakim było uruchomienie testowego programu, dostarczonego wraz z framework’iem TensorFlow.

python /usr/local/lib/python2.7/disst-packages/tensorflow/models/image/mnist/convolutional.py

Listing 8 Komenda uruchamiająca przykładowy program.

Program prawidłowo się uruchomił i testowa sieć uczyła się z domyślnego zbioru. Krok ten zakończył instalację środowiska.

## Inne wersje

TensorFlow jest elastycznym framework’iem. Umożliwia pracę na *python’ie 2.7–3.5*, z wykorzystaniem CPU lub GPU, jak również pracę na Linux’ie I Mac’u, (z wyłączniem GPU, dla systemów Mac OS X). Poniżej przedstawione komendy, które można zmienić w pierwszych krokach instalacyjnych (tylko dla GPU), z zaznaczeniem, że nie były testowane.

Python 3.4:

sudo apt-get install python3-pip python3-dev

Listing 9 Komenda instalacji python 3.

TensorFlow, dla Python 3.4

sudo pip3 install --upgrade https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow-0.8.0-cp34-cp34m-linux\_x86\_64.whl

Listing Komenda instalacji TensorFlow, dla python 3.4.

## Rozwiązywanie problemów

W tym miejscu znajdują się sugestie co do możliwych napotkanych problemów przy instalacji środowiska. Opisane zostały jedynie te problemy na które osoby pracujące nad projektem się natknęły.

* Pomimo prawidłowej instalacji CUDA nie jest wykrywana przez TensorFlow.

Tutaj należy zacząć od sprawdzenia czy prawidłowo dodane zostały informacje LD\_LIBRARY\_PATH i CUDA\_HOME. Może się tak zdarzyć, że nie. Należy wtedy jedną z dwóch komend wyświetlić zmienne środowiskowe i zweryfikować poprawność dodania.

printenv

echo $PATH

Listing 11 Komendy wyświetlające zmienne środowiskowe.

Pierwsza z nich wyświetla wszystkie zmienne środowiskowe, a druga tylko PATH. Ponowne dodanie zmiennych tym razem już z prawidłową lokalizacją powinno rozwiązać problem.

* Zmienne środowiskowe znikają po zamknięciu terminala.

Ten problem wymaga utrwalenia tych zmiennych poprzez dodanie do *~/.bashrc* informacji o tych zmiennych i ponowne wpisanie komendy.

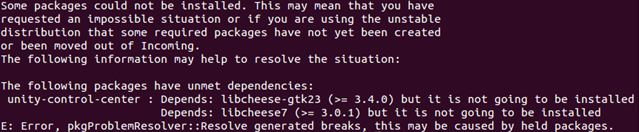
LD\_LIBRARY\_PATH=/usr/local/cuda/lib64:$LD\_LIBRARY\_PATH

Listing 12 Dodanie zmiennej środowiskowej.

Wcześniej prawdopodobnie zmienne nie zostały dodane do *~/.bashrc*.

* Problem z „zależnościami” przy instalacji CUDA.

Przykładowy komunikat:



Rysunek Problem z zależnościami przy instalacji plików CUDA.

Jest to informacja o problemie z samym systemem Linux, a nie z instalacją. Należy wtedy wykonać następującą komendę.

sudo apt-get -f install

Listing 13 Komenda naprawiająca błędne zależności.

Parametr –f oznacza naprawę. Komenda ta przeprowadzi próbę naprawy błędnych zależności. Następnie, dla pewności, można przeprowadzić prostą weryfikację, następującymi komendami.

sudo apt-get update

sudo apt-get upgrade

Listing 14 Komendy sprawdzające poprawność repozytorium i cache.

Te komendy sprawdzają czy repozytorium oraz cache są pozbawione błędów. To powinno rozwiązać problem.

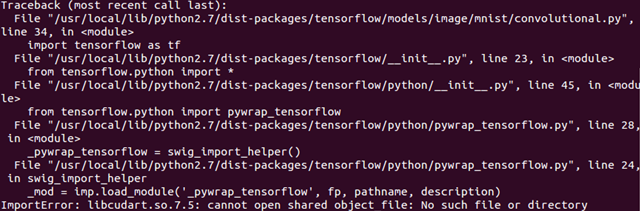
* Testowy program działa, ale wyświetla się o komunikat na początku o braku bibliotek cudNN.

Najprawdopodobniej wystąpił błąd przy instalacji. Należy sprawdzić lokalizację skopiowanych plików oraz zmienne środowiskowe czy prawidłowo CUDA została dopisana.

Nie należy się jednak przejmować tym problemem, ponieważ cudNN nie jest wymagane do poprawnego działania. Biblioteki te mają za zadanie m. in. przyspieszyć trenowanie sieci, więc można pominąć ten element środowiska, jeśli poprawienie lokalizacji skopiowanych plików nie pomaga.

* Problem z brakiem bibliotek CUDA przy próbie uruchomienia programu.

Możliwy błąd:



Rysunek Problem z importem bibliotek CUDA.

Przedostatni wiersz:

*ImportError: libcudart.so.7.5: cannot open shared object file: No such file or directory*

Oznacza to, że gdzieś został popełniony błąd przy instalacji bibliotek CUDA. Należny ponownie prześledzić proces instalacji w poszukiwaniu ewentualnych błędów.

# Rezultat prac nad projektem

# Instrukcja obsługi

# Podsumowanie

# Spis ilustracji, listingu i tabel

[Rysunek 1 Problem z zależnościami przy instalacji plików CUDA. 6](#_Toc449987939)

[Rysunek 2 Problem z importem bibliotek CUDA. 7](#_Toc449987940)

[Listing 1 Komenda instalująca sterownik graficzny. 3](#_Toc449987951)

[Listing 2 Komenda instalująca PIP. 4](#_Toc449987952)

[Listing 3 Komenda instalująca TensorFlow. 4](#_Toc449987953)

[Listing 4 Komendy potrzebne do instalacji CUDA. 4](#_Toc449987954)

[Listing 5 Komendy potrzebne do instalacji cudNN. 4](#_Toc449987955)

[Listing 6 Komenda ustawiająca zmienne środowiskowe. 5](#_Toc449987956)

[Listing 7 Weryfikacja poprawności instalacji TensorFlow poprzez import bibliotek. 5](#_Toc449987957)

[Listing 8 Komenda uruchamiająca przykładowy program. 5](#_Toc449987958)

[Listing 9 Komenda instalacji python 3. 5](#_Toc449987959)

[Listing 10 Komenda instalacji TensorFlow, dla python 3.4. 5](#_Toc449987960)

[Listing 11 Komendy wyświetlające zmienne środowiskowe. 6](#_Toc449987961)

[Listing 12 Dodanie zmiennej środowiskowej. 6](#_Toc449987962)

[Listing 13 Komenda naprawiająca błędne zależności. 6](#_Toc449987963)

[Listing 14 Komendy sprawdzające poprawność repozytorium i cache. 6](#_Toc449987964)

# Literatura

1. <https://en.wikipedia.org/wiki/Deep_learning> - podstawowe informacje o deeplearning’u. Stan na dzień 02.05.2016 r.
2. <https://www.tensorflow.org/> - strona główna framework’u, typu *open source*, TensorFlow. Stan na dzień 02.05.2016 r.
3. <http://keras.io/> - strona główna framework’u Keras. Stan na dzień 02.05.2016 r.
4. <https://developer.nvidia.com/cudnn> - strona główna bibliotek cudNN. Stan na dzień 02.05.2016 r.