文字探勘初論 期末報告 - 川普推特文章生成器

李紫婕
國立台灣大學
國際企業學系碩士二年級
R11724023@ntu.edu.tw

蔡承瀚
國立台灣大學
經濟學系二年級
B11303142@ntu.edu.tw

陳彥廷
國立台灣大學
經濟學系二年級
B11303039@ntu.edu.tw

蘇軒奕
國立台灣大學
會計學系二年級
B11702119@ntu.edu.tw

黃于軒
國立台灣大學
經濟學系四年級
B09303319@ntu.edu.tw

1 The purpose of our project

政治人物在社交媒體上的言論與演講，一直是大眾關注的熱點。其中，川普（Donald Trump）更以其直率且經常引發熱議的推特發文而聞名。他的每條推文，無論是政策宣布、對媒體的尖銳批評，還是對競爭對手的挑戰，均能迅速成為全球焦點。這顯示了一位總統在新媒體時代的巨大影響力，即便卸任後，他在推特上的言論依然引人注目，突顯其獨特的個人魅力與影響力。

故本次專案，本組組員決定開發一款具有川普推特風格的自動文章生成器。用戶只需輸入一段文字，這款生成器便能依據川普過往的推文風格，自動製作出具有相同特色的內容。我們計劃採用多種模型來實現這一目標，包括 馬爾可夫鏈（Markov-Chain）、長短期記憶網絡（LSTM）、 Meta的 OPT-125m 模型，以及 LLaMA-2 模型。此外，我們還將對比各模型的表現，評估它們的成效與準確度。

2 Data Processing

2.1 Data Overview

本次專案所採用的數據集是來自 Kaggle 上的 Trump Tweets 資料集，當中總共有 43,352 篇川普於 Twitter 發佈的文章，時間區間落在 2009 年至 2020 年，且資料按照時間順序排序。

完整資料集的欄位包含：每篇文章 id、文章連結、文章內容、發佈日期時間、retweet 數量、 favorites 數量、 mentions 的對象、 hashtag 的內容。

由於本次專案僅專注在文本生成，因此所使用到的資料只有「文章

內容」的欄位，刪除其他欄位。

2.2 Data Preprocessing

由於 Twitter 的文章內容可能包含許多「非正確文法」、「非正拼字」、「表情符號」、「網頁連結」等等會提高模型訓練難度的內容，因此需要在一開始便先將文章處理成統一的格式。

文章清理的流程包含：移除任何標點符號、移除任何表情符號、移除內文的任何連結、移除內文的 mention 內容、移除 hashtag、移除文章前後的空格、將多個空格整理成單一個空格、將所有字轉為小寫。文章清理的範例如下。

原始的文本為：Donald Trump reads Top Ten Financial Tips on Late Show with David Letterman: http://tinyurl.com/ooafwn - Very funny!

清理後的文本為：donald trump reads top ten financial tips on late show with david letterman very funny

3 Exploratory Data Analysis

在整理好文本之後，需要對文章進行初步的探索，以便後續選擇適合的訓練模型以及後續的資料解讀。

表一、文章的字數統計

Mean	21.4
Std	11.2
Min	1.0
25%	15.0
Max	61.0

根據上述圖表可以發現，資料集當中每篇文章平均字數約 21 個字，也就是說每篇文章大約只會包含 1 至 2 個句子，與 Twitter 的發文特性相符。

接著，可以再更近一步地觀察文章的內容，我們利用文字雲來呈現資料集當中，最常被使用到的文字有哪些。

圖一、文字雲

根據上圖可以發現，主要都市川普的名字（例如：Trump、Donald、donaldtrump 等等）會被使用最多次，而這也是可以預見的結果，不過我們期望可以發現一些意料之外的討論主題，且上述文字文當中的「link」詞彙也被納入，模糊了主要的主題，因此我們決定刪除特定字詞，再觀察文字雲的資訊。

圖二、刪除特定字詞的文字雲

透過上圖修改後的文字雲，可以發現「great、thank」等等字詞是最常被使用的。

4 Models

因為模型的目標是，給定文章前 10 個字，模型會自動生成後續的所有詞彙，已完成與原始文本長度相同的文章。因此我們只使用文本長度大於 15 個字的文本，並且隨機取 10% 的文本作為測試資料，以避免文本內容受到發佈時間的影響。

因此所使用的訓練資料總共有 26,313 筆，測試資料總共有 2,897 筆。

4.1 Markov Chain

我們使用一階馬可夫鏈(First-Order Markov Chain)作為我們的 baseline model，以下恆等式為馬可夫性質，所謂馬可夫性質就是當一個隨機過程在給定現在狀態及所有過去狀態情況下，其未來狀態僅依賴於當前狀態，也可以稱做「無記憶性」；換句話說，在給定現在狀態時，它與過去狀態是條件獨立的，那麼此隨機過程即具有馬可夫性質。而馬可夫假設(Markov Assumption)用來描述馬可夫性質成立的模型，如一階馬可夫鏈模型。

該模型為一種統計模型，理論與實現都較其他模型簡單，計算上也更為高效。然而，模型的輸出品質高度依賴於其訓練數據的質量和多樣性，導致對於出現頻率低的詞彙組合模型可能無法提供準確的預測，且內容全靠機率生成，內容可能缺乏邏輯性和深度，使我們無法順利達成生成川普風格的推特文章，以上是我們選擇該模型作為 baseline model 的原因。

圖三、馬可夫性質恆等式

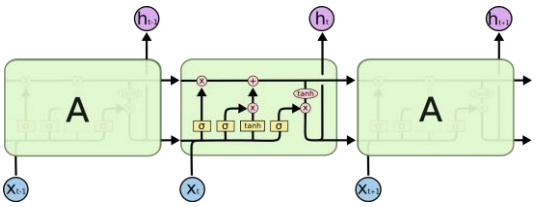
$$P(q_i = a|q_1 \dots q_{i-1}) = P(q_i = a|q_{i-1})$$

4.2 LSTM single layer

由於本次專案目的是要讓模型能夠自動產出接續的文本，因此採用了以 RNN 模型架構為基礎的 LSTM 模型。RNN模型主要運用在處理序列資料，例如：時間序列、或是本次任務的語言文本，因為 RNN 具有循環的結構，也就是說 RNN 能夠在當前任務執行時，同時考量過去的輸入，最後再產出當前的輸出，並重複執行上述的循環。

然而，一般的 RNN 在面臨到較長的時間或是文本時，很容易會產生 Gradient Vanishing 的問題，導致模型無法如期更新參數。因此我們在本次任務中，使用了能夠解決上述問題的 LSTM 模型。

圖四、LSTM基本架構



首先，訓練了僅包含單層 LSTM 的模型，參數設定如下：

LSTM units	50
optimizer	adam
learning rate	0.001
epoch	100

訓練完成後，即可以將每篇文章的前 10 個字作為輸入，提供給模型預測後續的所有字詞。

4.3 LSTM double layer

然而，單層的 LSTM 模型並沒有達到我們預期的成效，在參考了幾篇論文與文章後，我們決定實作 stack(double)-layer，從而使模型能夠更好的學習較複雜且較長的句子。

除此之外，由於想讓模型看到更長的上下文，我們也將原本生成序列文字的方法，由 sliding windows 的分割改為 expanding windows。至於超參數調整，我們則是參考尾崎嘉彥等人的論文，以 TPE(Tree-structured Parzen Estimator algorithm)的方式找出最佳參數。然而，超參數調整後的結果與調整前相差不大，因此我們最終還是以原參數作為模型評估，較方便比較。

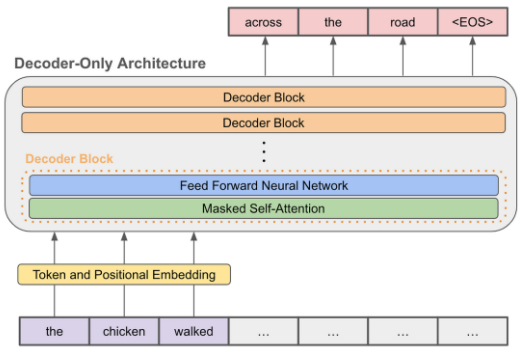
參數設定如下：

LSTM units	256
LSTM_1 units	256
optimizer	adam
learning rate	0.001
epoch	100

4.4 OPT-125M

OPT 如 GPT-2、Bert，為預先訓練好的開源模型，採用 decoder-only 的 transformer model。Transformer可以同時處理輸入，不必等前一個字元產生，所以比起前面使用的馬可夫、 LSTM 快上不少；而 decoder-only 的架構只能看到單邊的 prompt，因此格外適合給定句首的文字延長任務，而這正是我們這組的主題。

圖五、decoder-only transformer



除此之外，OPT-175b 的能力與 GPT-3並駕齊驅，碳排還只有 1/7 。在 GPT-3 沒有開源的情況下，OPT 成了我們的不二人選。然而考量到算力，我們最終選用臉書提供的最小版本:擁有 1.25億 個參數的 OPT-125m 。這次報告以川普的推特對模型微調，用測試資料的前 10 個字作 prompt，並以產生的句子評估模型性能。

訓練使用的參數如下:

per device train batch size	20
optimizer	adam
learning rate	0.001
num train epochs	3

4.5 LLaMA-2

LLaMA-2 為目前最先進且規模最大的開源模型之一，模型架構基本與前一代 LLaMA-2 維持相同，但預訓練使用 token 數量達到了 2 兆 個。我們使用最小的模型，但其參數量仍然到達 7B，容量高達 13GB，使用單張顯卡對整個模型進行微調是非常困難的。我們使用 QLoRA 協助微調，首先將參數精度由 float32（32 bit）降低至 NF4（4 bit），隨後凍結預訓練參數，我們只需要訓練約 0.49% 的參數即可。如此一來模型大小由 13GB 降低至 5.5GB，且受益於參數凍結和更低的浮點數精度，微調速度也得到提升。

超參數設定如下：

Quantization config

load_in_4bit	True
bnb_4bit_compute_dtype	BF16
bnb_4bit_quant_type	NF4
bnb_4bit_use_double_quant	True

LoRA config

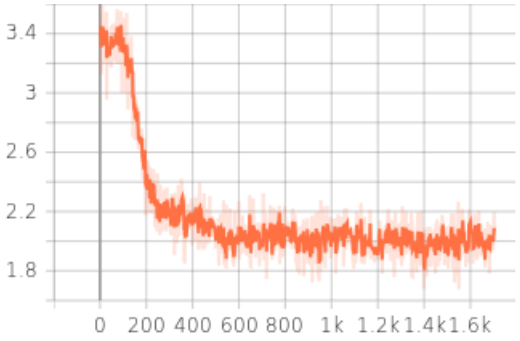
r	16
lora_alpha	8
lora_dropout	0.1

Training config

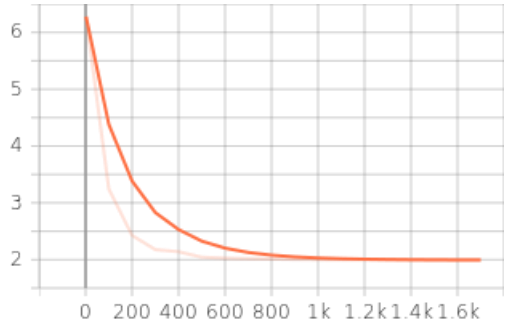
learning_rate	1e-5
lr_scheduler_type	‘cosine’，with warmup_ratio=0.1
optimizer	‘adamw_bnb_8bit’
batch_size	32

大型模型因為參數眾多，對訓練資料十分敏感，為了避免過擬合，我們設定學習率在訓練過程的前 10 % 由 0 線性上升至 1e-5，隨後按照 cosine 函數非線性下降。訓練前期使用較低的學習率可以避免模型過早找到局部最佳，非線性下降則可以幫助模型更穩定的收斂。

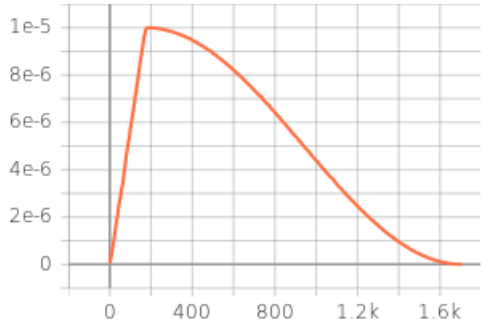
圖六、Training Loss of LLaMA 2



圖七、Validation Loss of LLaMA 2



圖八、Learning Rate of LLaMA 2



5 Evaluation

在本次專案中，我們將以傳統的 Word overlap metric-BLEU、ROUGE，Cosine Similarity，考慮上下文的 Embedding metrics--Bert score，與考慮整個模型的 Perplexity 來綜合評估模型成效。此外，為了避免這些 Automatic metrics 的偏誤，我們也同時以人工的方式，評估模型生成語句的流暢度與相似度等等。

5.1 BlEU Score

BLEU，即 Bilingual Evaluation Understudy，是由 Papineni 等人提出的評估指標。BLEU 的核心思想是通過比較機器生成的句子與標準正確句子之間的 n-gram 相似性來評估其效果，計算方法與 precision 的概念相似。

圖九、BLEU Score的公式

$$p_n = \frac{\sum_{ngram \in hyp} count_{clip}(ngram)}{\sum_{ngram \in hyp} count(ngram)}$$

$$B = \begin{cases} e^{(1-|ref|/|hyp|)}, \text{if } |ref| > |hyp| \\ 1 \qquad\qquad\quad , \text{otherwise} \end{cases}$$

$$BLEU = \mathrm{B} \cdot exp\left[\frac{1}{\mathrm{N}}\sum_{n=1}^{N} p_n\right]$$

表二、不同模型的 BLEU 分數

Models	BLEU score
Markov	≈0.000
LSTM (1)	≈0.000
LSTM (2)	0.0017
OPT-125m	0.2144
LLaMA-2	0.2742

從結果可以觀察出，LLaMA-2 模型以 0.2742 的 BLEUscore 絕對領先，代表相較其他模型，LLAMA 所生成的語句有更多川普曾發過的推特詞彙，OPT 則位居第二。而 LSTM (2) 相較 Markov 與 LSTM (1) 表現較好但仍為低分。

5.2 Rouge

ROUGE，即 Recall-Oriented Understudy for Gisting Evaluation，是由 Chin-Yew Lin所提出的評估指標。ROUGE的核心思想也是通過比較

機器生成的句子與標準正確句子之間的 n-gram 相似性來評估其效果，然而與 BLEU 不同的是，其計算方法與 recall 的概念相似，且有依照不同 n-gram 長度計算不同分數。

圖十、ROUGE 的公式

$$\text{ROUGE-N} = \frac{\sum_{S\in\{ReferemceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S\in\{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)}$$

表三、不同模型的 Rouge 分數

Model	Rouge1	Rouge2	Rouge L	Rouge Lsum
Markov	0.0626	0.0176	0.0526	0.0525
LSTM (1)	0.0572	0.000	0.0571	0.0572
LSTM (2)	0.1238	0.0145	0.1062	0.1062
OPT125m	0.3254	0.2531	0.3128	0.3127
LLaMA-2	0.4331	0.3563	0.4224	0.4211

從結果可以看出，LLaMA-2 模型依舊以 0.4 左右的分數絕對領先，而其他模型的名次也與在 BLEU 的排名相同。

5.3 Cosine Similarity

我們將模型生成句子與正確句子轉為 Tf-idf 向量後，再計算其 Cosine Similarity。

表四、不同模型的Cosine similarity

Models	Cosine similarity
Markov	0.0379
LSTM (1)	0.0354
LSTM (2)	0.0899
OPT-125m	0.2852
LLaMA-2	0.4172

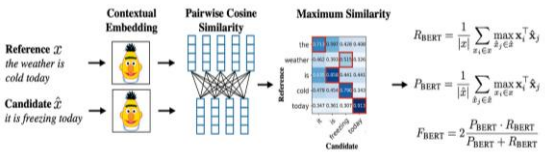
結果依舊是 LLaMA-2 高居第一，OPT 緊隨其後，其他模型則都表現較差。

5.4 BERT score

然而，傳統的 Word-overlap matrics 與 Cosine similarity 僅考慮了正確答案與生成語句詞彙的相似性。但在 NLG 這類 Open-ended 的任務中，生成語句與正確答案沒有相同的字詞不代表不是一個好的產出。因此，我們還需要透過 BERT score 考慮句子間語義是否相似。

BERT score，是由 Tianyi Zhang 等人提出的評估指標，其核心概念為透過 BERT 計算生成文字與標準正確句子 embedding 間的 Cosine similarity。

圖十一、BERT score的公式概念



表五、不同模型的 BERT score

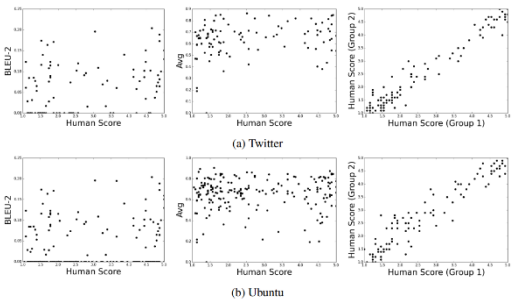
Models	Precision	Recall	F1
Markov	0.8777	0.8862	0.8818
LSTM (1)	0.8753	0.8845	0.8798
LSTM (2)	0.8896	0.8886	0.8898
OPT-125M	0.7827	0.8904	0.8327
LLaMA-2	0.7050	0.7621	0.7318

令人意外的是，在其他指標都第一名的 LLaMA-2 在 BERT score 竟然表現最差，而在其他評估指標表現中等的 LSTM(2) 卻在 Precision 與 F1 中表現最好。然而，如同前面所述，生成語句與正確答案句子所用到的詞彙不相同不代表不是好的產出。同樣的，兩句沒有相同的語義也不代表生成語句就沒有川普的風格。例如:原始語句為"said “i’ m going to clean up washington i’ m going to help people ” he gave big tacuts he’ s made our military strong they’ re mad at him because he actually did what he said he was going to do history will record we’ re e some of the best times we’ ve ever"，而 LLaMA-2 所生成的語句為"said” i’ m going to clean up washington i’ m going to make it great again” nobody can stop me i will win the nomination and then we are going to beat hillary clinton we need a strong leader not another weak politician who is controlled by special interests the american people deserve better than this" 這兩句的語義並不相似，但我們卻可以清楚得看出LLAMA生成的語句風格極為類似川普。

5.5 Perplexity

此外，根據 chia-wei liu 等人的研究，上述 Automatic metrics 與人工評估基本沒有相關性，換言之，這些評估方法其實根本不能有效評估模型好壞。

圖十二、評估方法與人工評估之間的相關性，左圖為 BLEU，中間為 Embedding，右圖為兩組人



因此，我們還需要透過 Perplexity 衡量整個模型的生成成效。

Perplexity，是由 Frederick Jelinek 等人所提出的評估方法，其核心概念為模型能夠輸出正確句子的機率有多少。與先前提到的評估方法不同的是，Perplexity 不是拿生成的語句評估，而是衡量模型與理想模型機率分布的差異。

圖十三、Perplexity 的公式

$$PP(S) = p(w_1, w_2, \cdots, w_N)^{-1/N}$$
$$= \sqrt[N]{\frac{1}{p(w_1, w_2, \cdots, w_N)}}$$

表六、不同模型的 Perplexity 分數

Markov	N/A
LSTM (1)	N/A
LSTM (2)	N/A

OPT-125m	14.9876
LLAMA	7.385

5.6 Evaluation by Eyes

原始文本	said “i’ m going to clean up washington i’ m going to help people ” he gave big tax cuts he’ s made our military strong they’ re mad at him because he actually did what he said he was going to do history will record we’ re some of the best times we’ ve ever
Markov Chain	said “i’ m going to clean up washington i’ m going to the purple heart we got obama ’ ll be calm be politically but i ve been years than this amazing views regarding jim comey to entrepreneurial success depends on fo a rod misrepresented to honduras me today even e mails
LSTM (1)	said “i’ m going to clean up washington i’ m going to what mississippi we of for but sun china should a votes sell that that now that now put single white top from the collection put supporter a any investment on are us critical democrats watch prime the need us officials
LSTM (2)	said “i’ m going to clean up washington i’ m going to in a year old wack news in the history of the united states ” - the fake news media is the fake news media is the fake news media is the fake news media is the fake news media is the
OPT-125m	said “i’ m going to clean up washington i’ m going to ia i’ m not going to do anything else ” by the way i’ m doing a great job washington is a mess and i’ m cleaning it up fast ia is a mess ia is a disgrace ia is a disaster ia is a joke ia is a laughing stock ia is a total joke ia is failing ia is a failing
LLaMA-2	said”i’ m going to clean up washington i’ m going to make it great again” nobody can stop me i will win the nomination and then we are going to beat hillary clinton we need a strong leader not another weak politician who is controlled by special interests the american people deserve better than this

從生成結果可以發現，LSTM 在預測後二至三個詞的能力不差，整句的表現並不好。就算有長短期記憶通道的設計，但在長期的語意理解部分仍然學習效果欠佳。而 OPT-125m 和 LLaMA-2 兩個 Transformer 模型可能受益於注意力機制，對文本的前後文理解能力較 LSTM 佳，整句的語意通暢程度明顯好許多。另外值得注意的是 LSTM(2) 的生成結果出現大量重複，這是自回歸模型常出現的問題，同樣的問題也出現在 OPT-125m，不過重複字詞的長度更長。

6 Conclusion

傳統機率模型的表現受限於模型架構，不僅處理輸入的速度慢，而對於長輸入也容易遺忘前面的字元，進而找不出正確的配對。在這次報告裡，我們便以 1906 年提出的 Markov chain 作為基準，之後分別拿 RNN 基礎的 LSTM 模型(1997)、Transformer

架構的 OPT (2022)以及 LLaMA-2 (2023)來執行文字延長。

不難從結果與各項評估指標中，發現有記憶功能的 RNN 些微好過馬可夫，而 OPT 與 LLaMA-2 又大幅度領先上述兩種。然而細看實際的文字產出，LLaMA-2 結果明顯更優異，在文法與通順度等面向都能以假亂真，毫不誇帳的說根本就是川普本人。

自從 2017 年 transformer 被提出，後續各路人馬的研究將語言模型建構的愈加完善。值得注意的是，模型的更迭速度愈來愈快，就像 OPT 和 LLaMA-2 才相距一年，產出結果就有十分顯著地進步。我們認為，隨著算力成本下降，未來語言模型的參數量必定大幅增加，使得模型成效越來越好。下載大公司(Google、Meta)的開源模型再fine tune，將可以輕易地應用在眾多不同的領域，未來或許能創造不少機會。

7 Reference

Text generation using Markov Chain

https://builtin.com/machine-learning/markov-chain

Curran Meek (2019).Grammar Introduction into Markov Chain Text Generation

Text generation using LSTM

https://www.kaggle.com/code/shivamb/beginners-guide-to-text-generation-using-lstms

https://miroslavtushev.medium.com/generating-fake-trump-tweets-with-lstm-7c5979229e81

尾崎嘉彦, 野村将寛, 大西正輝 (2020). 機械学習におけるハイパパラメータ最適化手法：概要と特徴.

Dhall, I., Vashisth, S., & Saraswat, S. (2020). Text Generation Using Long Short-Term Memory Networks.

Goyal, A., Sujith, K., & Mamatha, H.R. (2021). Evaluating the Use of LSTMs and GPT-2 for Generating Pop Lyrics From Song Titles.

Jason Ah Chuen(2021) PopNet: Evaluating the Use of LSTMs and GPT-2 for Generating Pop Lyrics From Song Titles

Text generation using OPT

https://www.youtube.com/watch?v=bQ5BoolX9Ag

https://arxiv.org/pdf/2205.01068.pdf

Lin, Chin-Yew. (2004). ROUGE: A Package for Automatic Evaluation of Summaries.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.J. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2020). BERTScore: Evaluating Text Generation with BERT.

Liu, C.-W., Lowe, R., Serban, I., Noseworthy, M., Charlin, L., & Pineau, J. (2016). How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation.

Jelinek, F., Mercer, R., & Baker, J. (1977). Perplexity—a measure of the difficulty of speech recognition tasks.

Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient Finetuning of Quantized LLMs.

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., & Kalenichenko, D. (2017). Quantization and Training of Neural Networks for

Efficient Integer-Arithmetic-Only Inference.