

## PNASNet-5 および Wide&Deep Mix アプローチを用いた バナー広告の CTR 分類

### Click Through Rate based Banner Ads Classification using PNASNet-5 with Wide and Deep Mix modeling approach

平岩 篤信<sup>†</sup> 渡邊 太郎<sup>†</sup> ハサン マハモドウル<sup>†</sup> グエン・ディン・マイン<sup>†</sup> ディン・チョン・タン<sup>†</sup>  
グエン・ディン・タオ<sup>†</sup> 神谷 寛<sup>†</sup> 大西 一貫<sup>†</sup>  
Atsunobu Hiraiwa<sup>†</sup> Taro Watanabe<sup>†</sup> Hassan Mahmodul<sup>†</sup> Nguyen Dinh Manh<sup>†</sup> Dinh Trong Thang<sup>†</sup>  
Nguyen Dinh Tao<sup>†</sup> Hiroshi Kamiya<sup>†</sup> Kazuhiro Ohnishi<sup>†</sup>

#### 1. はじめに

インターネット広告のバナー画像における広告クリエイティブとしての画像評価を行う際、その評価指標として CTR (Click Through Rate) が広く用いられているため、機械学習の応用研究として CTR を直接予測するもの[1-2]や CTR 期待値を基準に品質の程度をクラスとして分類を行うものがある。[3]では識別率 69%を超える結果が報告されているが、社内の経験を積んだデザイナーの分類性能が 82%以上であるため、それと同等以上の性能が要求されており、さらなる識別率向上が期待されている。人間の分類性能以上のバナー広告の CTR 分類を実現することにより、CTR が大きい効果的な広告と CTR が小さい非効果的な広告に分類するシステムを提案する。1) 広告を配信した結果観測される日時やその 24 時間における表示回数などの集計された情報である配信実績情報、2) その広告がどんな性別のどんな年代でどの地域に配信されたかを表すターゲット情報、3) 画像から学習済みモデルを使用した 4320 次元の特徴ベクトル、これら 3 つの情報を入力とする 7 層の隠れ層からなる深層学習モデルにより学習し、新しく作成したバナー広告画像と配信設定、および想定される実績 (コストなど) を基にその画像が、CTR が大きい効果的なバナー広告画像と CTR が小さい非効果的なバナー広告画像のどちらに属しているかを出力する。従来、デザイナーがそれぞれ独自に判断し、時間をかけて行われてきたバナー広告画像の効果予測を、システムからデータに基づいた予測として提供されるようになる。これにより、デザイナーの効果予測を行う時間が低減し、より製作に集中できるようになり、熟練デザイナーになるためにはまだ経験値が必要な若手デザイナーも同じ画像かつ同じ条件であればその時々で変化することなく安定した効果予測を得ることができる。実業務で運用された実績データを基に学習したモデルが人間の分解能である 82%を超えることにより、実務上有効であることを確認するため、自社で運営している 2 媒体について、媒体ごとに実際に配信した結果得られる実績情報を元にモデリングを行い、その識別率の評価を行う。

2. では CTR 分類に対する高精度化のニーズ及び画像が高解像度することによる影響について述べ、それらを解決するためのアプローチについて、3. にて PNASNet-5 および Wide & Deep Mix アプローチを用いたネットワーク構造とそれを利用したバナー広告の CTR 分類システムの入出力、および実際の画面イメージについて紹介する。4. では自社で運営している 2 媒体について実際に配信を行って観測

された広告実績情報を用いたバナー広告画像の識別率評価試験について紹介し、5. にて評価試験結果を述べる。

#### 2. 広告業界におけるバナー広告の CTR 分類に対する高精度化のニーズと画像の高解像度化問題

スマートフォンを始めとしたポータルデバイスでバナー広告はインターネットユーザとの接触回数や時間が増大しており、そのコンテンツのライフサイクルは年々短くなってきている。これに伴い、従来行われてきたデザイナーによる広告パターン作成と吟味する時間の確保が困難となってきたことから、より効果的なバナー広告画像評価システムが求められている。

インターネット広告のバナー画像における広告クリエイティブとしての画像評価を行う際、その評価指標として CTR (Click Through Rate) が広く用いられているため、機械学習の応用研究として CTR を直接予測するものや CTR 期待値を基準に品質の程度をクラスとして分類を行う CTR 分類問題がある。バナー広告 CTR 分類問題の最新研究[3]では、識別率 69%を超える結果が得られているが、ビジネス活用を検討した際に業務担当者がその結果に対して確信するには、社内の経験を積んだデザイナーの識別率が 82%以上であるため、より高い精度での判定が必要となっている。またレコメンデーションシステム分野では、線形モデルと深層学習モデルを融合した Wide & deep model[4]チュートリアルでの年収 2 値分類予測システム[5]は、83%を超える成果が確認できるが、画像を含むデータを入力した応用には研究の余地がある。画像を扱った転移学習[6-7]では、事前学習モデルに、3 層程度の隠れ層を含む完全結合ネットワークを追加し、再学習する場合が多い。[8]では、8 層の隠れ層を含む完全結合ネットワークも出現しているものの、十分な適応結果を得るにはさらなる技術的ブレイクスルーが必要である。また、最近のバナー広告は、端末の高精細化、通信コストの大幅な低下により、高解像化が進んでおり、このニーズに適応した事前学習モデルの公開が切望されている。事前学習モデル pnasnet\_large[9]は、ImageNet で、90%以上の検索性能があり、331x331 の解像度を入力として作成されており、現在のバナー広告へのニーズを満たすものである。

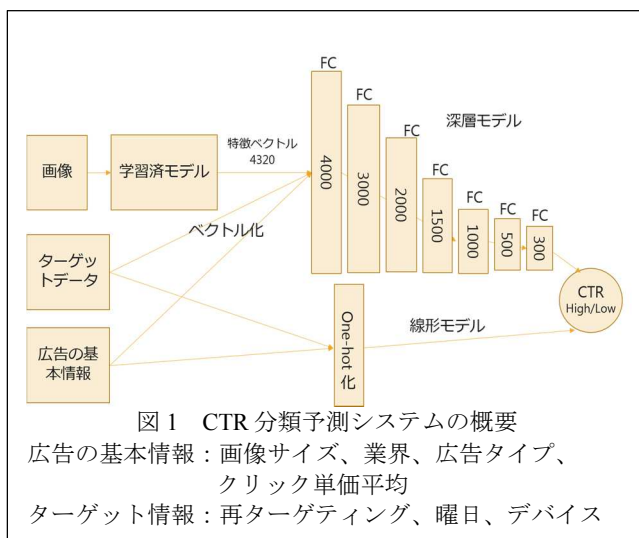
#### 3. PNASNet-5 および Wide & Deep Mix アプローチを用いたバナー広告の CTR 分類

PNASNet-5 および Wide & Deep Mix アプローチを用いたバナー広告の CTR 分類システムのネットワーク構造を 1 図に示す。システムは広告画像データ、その広告画像を配信するターゲット情報 (性別、年齢、地域)、およびその設

定によって配信された日時とその時の配信実績情報（表示回数[impression]、配信で発生したコスト[cost]、日予算[daily budget]など）を入力とし、入力された配信設定下におけるその画像の配信時点でのCTRが、過去に配信された広告画像をCTRの降順に並べたものの中で上位である確率、すなわちその画像が効果的なバナー広告画像である確率を出力する。画像データ部分は331x331にリサイズし、ImageNetで事前学習したモデル pnasnet\_large[9]から生成される4320次元からなる中間出力を特徴ベクトルとして後続のネットワークの入力として使用する。ターゲット情報（属性データ）はカテゴリカル変数であるため、ワンホットベクトル化して入力情報とする。広告配信実績情報は数値情報のため、そのまま列挙したものをベクトルとする。Wide & deep model[4]の深層部分は、バナー広告への適応化への必要性から下記で定義される隠れ層が7層から成る完全結合ネットワークである。

隠れ層は入力に近いものから順に、4000、3000、2000、1500、1000、500、300個のノードによって構成されている。

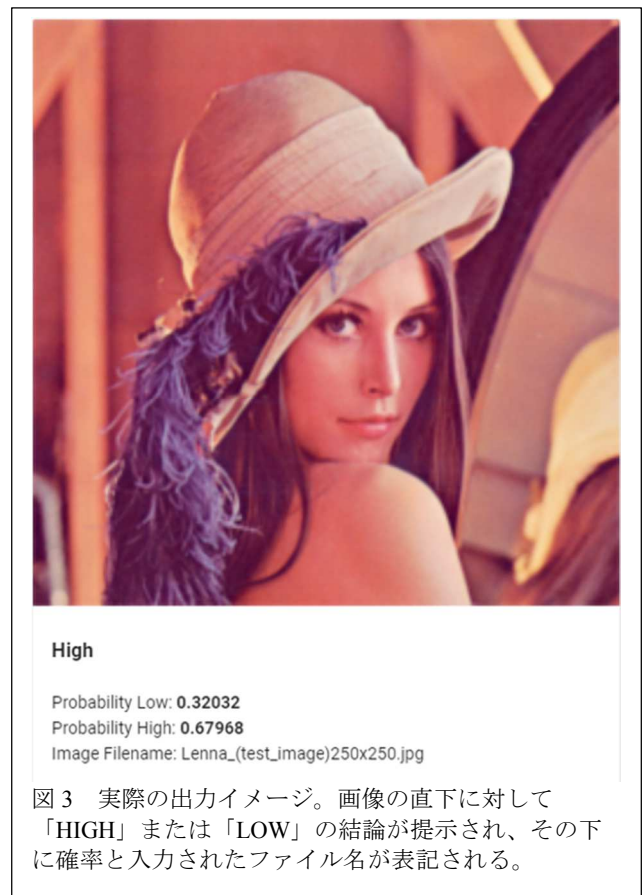
学習工程には使用時の前の月から始まる直近3か月の実際に配信された配信実績情報とその時使用されたバナー広告画像群を使用してモデルを作成し、デザイナーは最新のモデルを読み込んで曜日などの配信設定や想定する業種など評価したい画像と合わせて入力して利用する。（図2）システムは入力された未配信の画像を、配信設定を基にCTR分類を行い、未配信の画像を実際に配信する前に効果的なバナー広告画像か、非効果的なバナー広告画像かを予測する。モデルの最終出力では、効果的なバナー広告画像である確率が出力されるため、ユーザインタフェースでは出力が0.5より大きい値である時に「効果的なバナー広告画像である」ことを意味する「HIGH」という結論を出力し、それ以外の時は「非効果的なバナー広告画像である」ことを表す「LOW」の結論を出力する。（図3）画像ファイルは複数と同時に入力することができるが、内部では1枚ずつを評価し出力したものを画面上でまとめている。システムがCTRを基準に効果的なバナー広告画像か非効果的なバナー広告画像かを出力するため、デザイナーは実際に広告としてその画像を配信する前に複数のデザイン候補を作成し、それらの画像群のうちどれが効果的かどうかを評価することができる。



#### 4. 実際の配信実績情報とバナー広告画像を用いたCTR分類における識別率評価試験

提案システムが1) デザイナーの効果予測を行う時間を低減する、2) 若手デザイナーも同じ画像かつ同じ条件であれば安定した効果予測を実業務で活用できる、の2つを達成するためには、実運用された実績データを基に学習したモデルが人間の分解能である82%を超えていることが重要である。モデリングの識別率が人間のそれを超えていることを確認するため、自社で運営している主要な2媒体であるGoogle Adsのうち、ネットワークが「Display」であるGoogle ディスプレイネットワーク（以下、GDN）とYahoo! ディスプレイアドネットワーク（以下、YDN）について、媒体ごとに実際に配信した結果得られる実績情報を基にモ

図2 入力画面イメージ



デリグを行い、2つのモデルのそれぞれの識別率の評価を行う。

#### 4.1 データセット：実績情報の取得期間と広告の範囲

実験にはデータセットとして、GDNの実績データとYDNの実績データを使用する。GDNの実績データは2019年3月1日から2019年5月31日までの3ヶ月で1日毎の集計値である。YDNの実績データは、2019年3月1日から2019年5月31日までの3ヶ月で1時間毎の集計値である。このデータには、顧客ID、日付、配信対象となるデバイス(パソコン、スマートフォン、タブレット)、画像サイズ、画像を表すクリエイティブID、クリック単価の平均、クリック率(CTR)、リターゲティング情報(サイトを訪問したが購入に至らなかった人たちをターゲットとする広告)が含まれる。

2015年から登場したユニバーサルアプリケーション(UAC広告)やインフィード広告などは純粋なバナー広告という観点で適していないため、実績データの中には含まれない。また、動画広告も含まず、静止画で構成されたバナー広告を対象としており、gifアニメーションなどで構成された画像ファイルは最初の1枚目のフレームのみを静止画として変換し対象としているが、このケースは全体から見て1%未満であるため、大きな影響はない。

#### 4.2 前処理：外れ値の定義と除外と集計の粒度調整

実績データは媒体から直接取得した状態のままでは重複が含まれており、ノイズとなるため、広告配信実績情報、ターゲット情報の非数値データをFeature Crossでハッシュ処理、Embeddingで、ベクトル化したデータと画像を作成する。CTRが0と欠落データのある実績データは、意味がないと考え削除している。また、表示回数が5以下ではサンプルが少なすぎて極端なCTRの数値となりやすいため、除外対象としている。実績データには日付情報のみが記録されているが、ビジネスロジックには曜日の概念が大きく反映されているため、日付から曜日を求め内部パラメータとして使用している。また、自社が独自に管理

しているデータベースから顧客IDを用いてその顧客の業種と広告タイプを得ている。時間分解能は、曜日を使用したので、1日の平均を用いることとし、YDN側のデータは毎時データとなっているため、各実績データの項目については再集計を行い、1日単位の時間粒度で実績データが揃った状態とする。

この前処理を実施した結果、実験に使えるデータ量はGDNの実績データが35,158件、YDNの実績データが152,220件となった。

#### 4.3 学習、テストデータ

学習データは、前処理された実績データの8割を使用し、残り2割をテストデータとして乱数で振り分けたものを用いる。このテストデータのシャッフルをPandasで行っているコードを下記に示す。

```
df.sample(frac=1).reset_index(drop=True)
```

各媒体のデータの数はそれぞれ、GDNが学習データ数28,148、テストデータ数7,037であり、YDNが学習データ数121,724、テストデータ数30,496である。CTR 2値分類の閾値は、Pandas describeの50%：中央値(median)で求め、GDNの閾値が0.501765、YDNの閾値が2.4390244である。

#### 4.4 実験のパラメータ

最適化には、ProximalAdagradOptimizer、L1正則化を0.0001、L2正則化0.0001、学習率0.0005、バッチサイズ128(YDN)、64(GDN)を用いた。

### 5. GDN データセットと YDN データセットを用いた CTR 分類器の識別率評価

#### 5.1 GDN データセットにおける識別率の評価

GDNの実績データに対して、図4に示すとおり、データが最もよくなった時点のエポック数は26(重み係数の更新回数が11,400K付近)であり、その結果はauc:0.873を確認する。なお、このときtensorboard上に表示されるスコアはそれぞれ、accuracy:0.793、auc\_precision\_recall:0.875の値を得る。また、

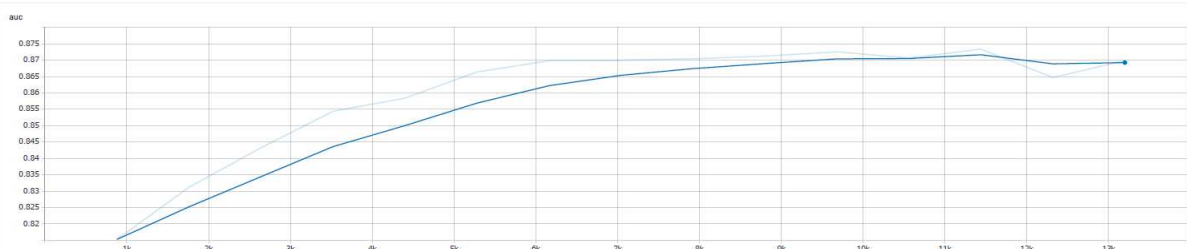


図4 GDN データセットにおける学習の推移 横軸は重み係数の更新回数、縦軸は auc の値を示す

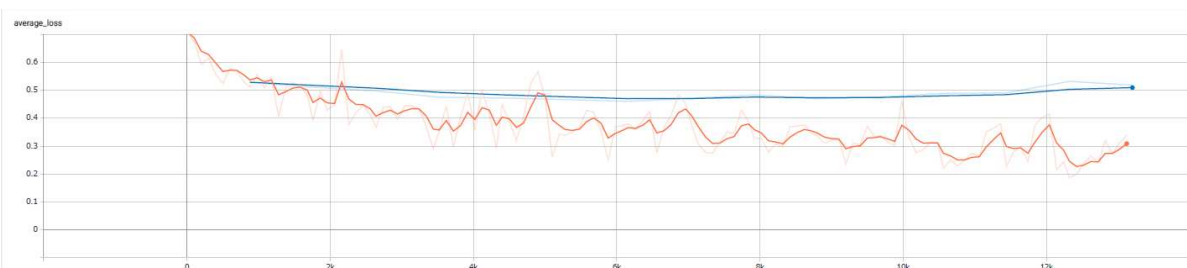


図5 GDN データセットにおける学習の推移 横軸は重み係数の更新回数、縦軸は average\_loss の値を示す 青い線はテストデータにおける値であり、オレンジの線は学習中の値である

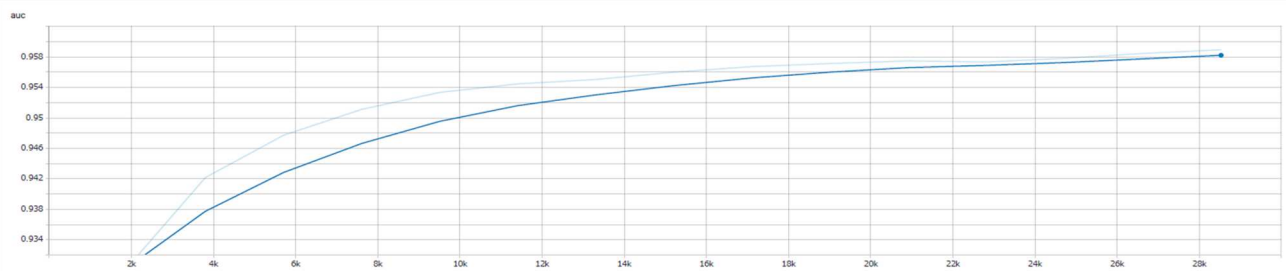


図 6 YDN データセットにおける学習の推移 横軸は、重み係数の更新回数、縦軸は、auc の値を示す

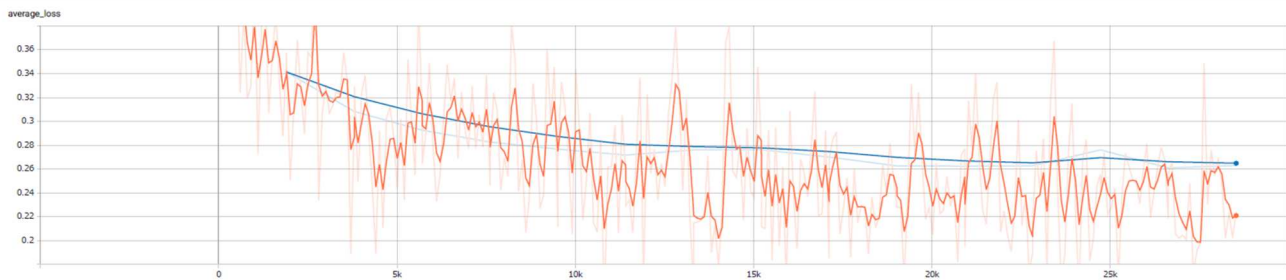


図 7 YDN データセットにおける学習の推移 横軸は、重み係数の更新回数、縦軸は、average\_loss の値を示す  
青い線はテストデータにおける値であり、オレンジの線は学習中の値である

図 5 に示すとおり、11,400K ではテストデータにおける average\_loss の値と学習中における average\_loss の値はさほど乖離していないが、それ以上の更新回数では乖離が大きくなり始めているため、過学習の傾向がある。そのため、過学習ではない状態での auc は 0.873 が最高であり、システムとして有効に機能するスコアであることを確認する。

## 5.2 YDN データセットにおける識別率

YDN の実績データに対して、図 6 に示すとおり、データが最もよくなった時点のエポック数は 30 (グラフ上の最右部) であり、その結果は、auc : 0.959 を得る。GDN の場合と同様に、このとき tensorboard 上に表示されるスコアはそれぞれ、accuracy: 0.889、auc\_precision\_recall : 0.965 である。

図 7 で確認できるように、テストデータでのロスと学習中のロスはほとんど乖離しておらず、過学習していないことを確認する。また、性能がすでに人間の識別率である 0.82 を大きく超えていることから、十分に有効な識別器として機能することを確認する。

## 6. おわりに

自社で運営している主要な 2 媒体について媒体ごとに実際に配信した結果得られる実績情報を元にモデリングを行い、識別率の評価実験を行った結果、GDN では過学習ではない状態での auc は 0.873 (すなわち 87.3%) が最高であり、YDN では auc : 0.959 (95.9%) を得る。これは、経験のある社内の人間の識別率の 82% を超えており、実業務で活用するに十分有効であることを確認する。

現在、社内システムとして業務フローを整理し、導入を進めている。

## 参考文献

- [1] Junxuan Chen, Baigui Sun, Hao Li, Hongtao Lu, Xian-Sheng Hua: Deep CTR Prediction in Display Advertising, <https://arxiv.org/pdf/1609.06018.pdf>, (2016).
- [2] 岩崎祐貴: 深層学習による Facebook 広告の CTR 予測, 人工知能学会全国大会, 4Pin1-14 (2018).
- [3] Nicolas Michel, Hayato Sakata, Keita Kurita, Toshihiro Yamasaki: Banner Click Through Rate Classification Using Deep Convolutional Neural Network, 人工知能学会全国大会, 101-01 (2018).
- [4] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, Hemal Shah: Wide & Deep Learning for Recommender Systems, <https://arxiv.org/pdf/1606.07792.pdf>, (2016).
- [5] [https://github.com/tensorflow/models/tree/master/official/wide\\_deep](https://github.com/tensorflow/models/tree/master/official/wide_deep)
- [6] [http://cs231n.stanford.edu/slides/2017/cs231n\\_2017\\_lecture7.pdf](http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture7.pdf)
- [7] [https://www.tensorflow.org/beta/tutorials/images/transfer\\_learning](https://www.tensorflow.org/beta/tutorials/images/transfer_learning)
- [8] Tero Karras, Samuli Laine, Timo Aila: A Style-Based Generator Architecture for Generative Adversarial Networks, <https://arxiv.org/pdf/1812.04948.pdf>, (2019).
- [9] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy: Progressive Neural Architecture Search, <https://arxiv.org/pdf/1712.00559.pdf>, (2018).