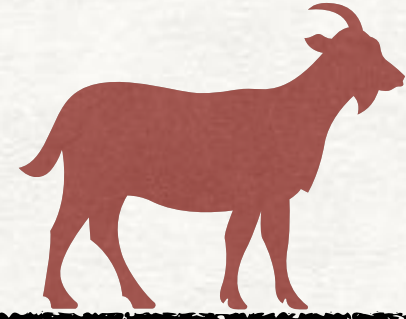




Deep Learningを用いた 競馬の勝ち馬予測プログラムの作成



日本大学文理学部情報科学科 谷研究室
宮脇 祐太



Deep Learn

目次

0：機械学習・ディープラーニングについて

1：はじめに

1 - 1：競馬とは

1 - 2：競馬における機械学習の活用事例

1 - 3：競馬予想アルゴリズムを競う大会

1 - 4：演習内容・目的

2：演習方法

2 - 1 競馬予想プログラムの作成する手順

2 - 2 webページから情報を取得

2 - 3 データの前処理・整形

2 - 4 DNNモデルの構築

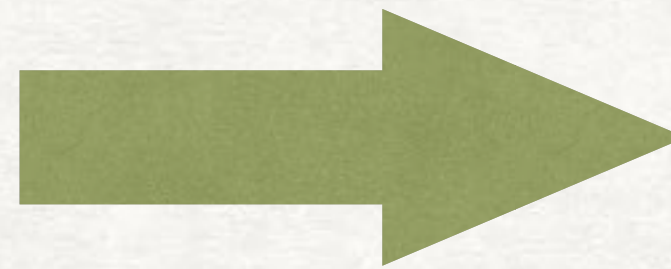
3：結果考察

0 機械学習・ディープラーニングとは

機械学習

「明示的にプログラムしなくても学習する能力を
コンピュータに与える研究分野」 (アーサー・サミュエル 1959年)

大量のデータ



未知のデータ
に対する答えを予測

0 機械学習・ディープラーニングとは

ディープラーニング (Deep Learning)

「多くの層を持ったニューラルネットワークを用いた
機械学習の総称」 (引用) 「深層学習」 人工知能学会監修, 第4版2016年

画像認識

囲碁・将棋

自動運転

株取引

0 機械学習・ディープラーニングとは

ニューラルネットワーク

「脳神経系をモデルにした情報処理システムのこと」

(引用) 村上研究室 ニューラルネットワーク - 画像処理・理解研究室 - 愛媛大学

<http://ipr20.cs.ehime-u.ac.jp/column/neural/index.html> (参照日時:2017/12/12)

特に、層の深いニューラルネットワークモデルを
DNN(Deep Neural Network・ディープニューラルネットワーク)と呼ぶ

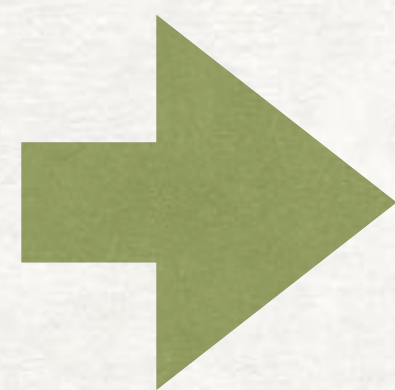
0 機械学習・ディープラーニングとは

ディープラーニングの動作

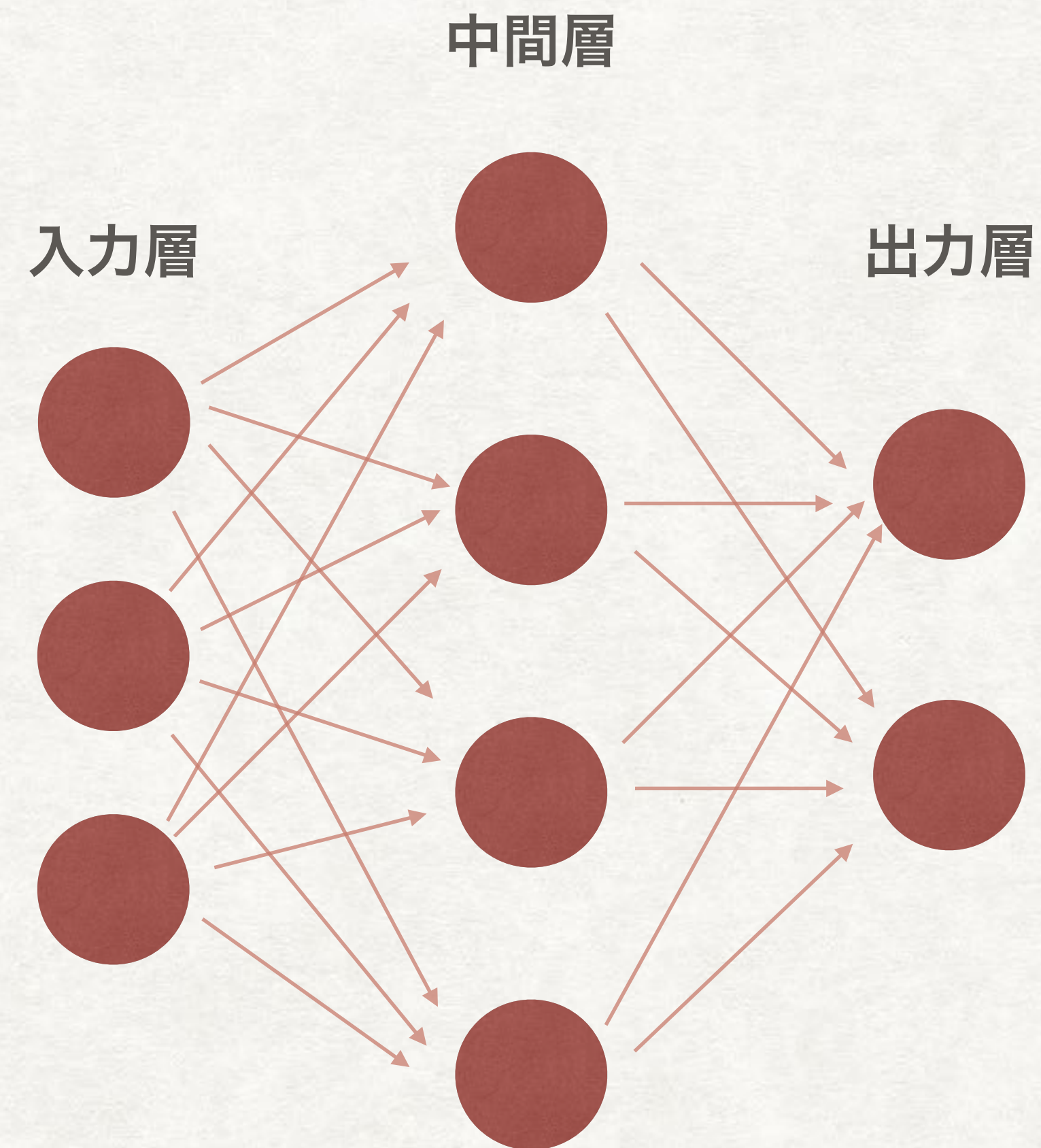
ラベル付きデータを入力とし
出力値を計算

ラベルと出力値の誤差を計算

重みを修正



ラベルと出力値の
誤差を小さくなる様
重みを修正



0 機械学習・ディープラーニングとは

ディープラーニングの動作

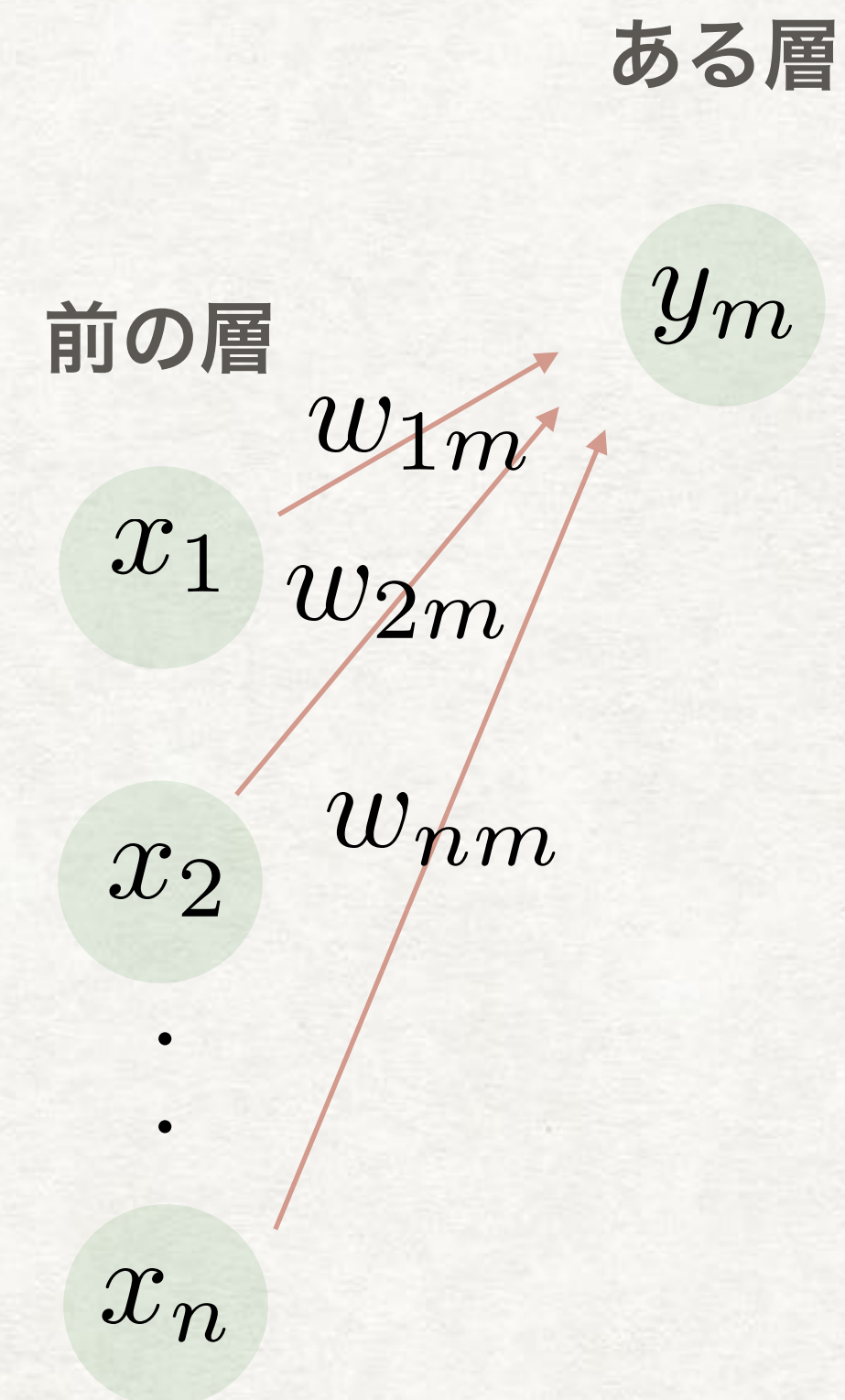
1. ラベル付きデータを入力
とし出力値を計算

$$y_m = \sum x_n \cdot w_{nm}$$

y_m : ある層 y の m 番目のノードの値

x_n : 前層の n 番目のノードの値

w_{nm} : x_n と y_m 間の重み



0 機械学習・ディープラーニングとは

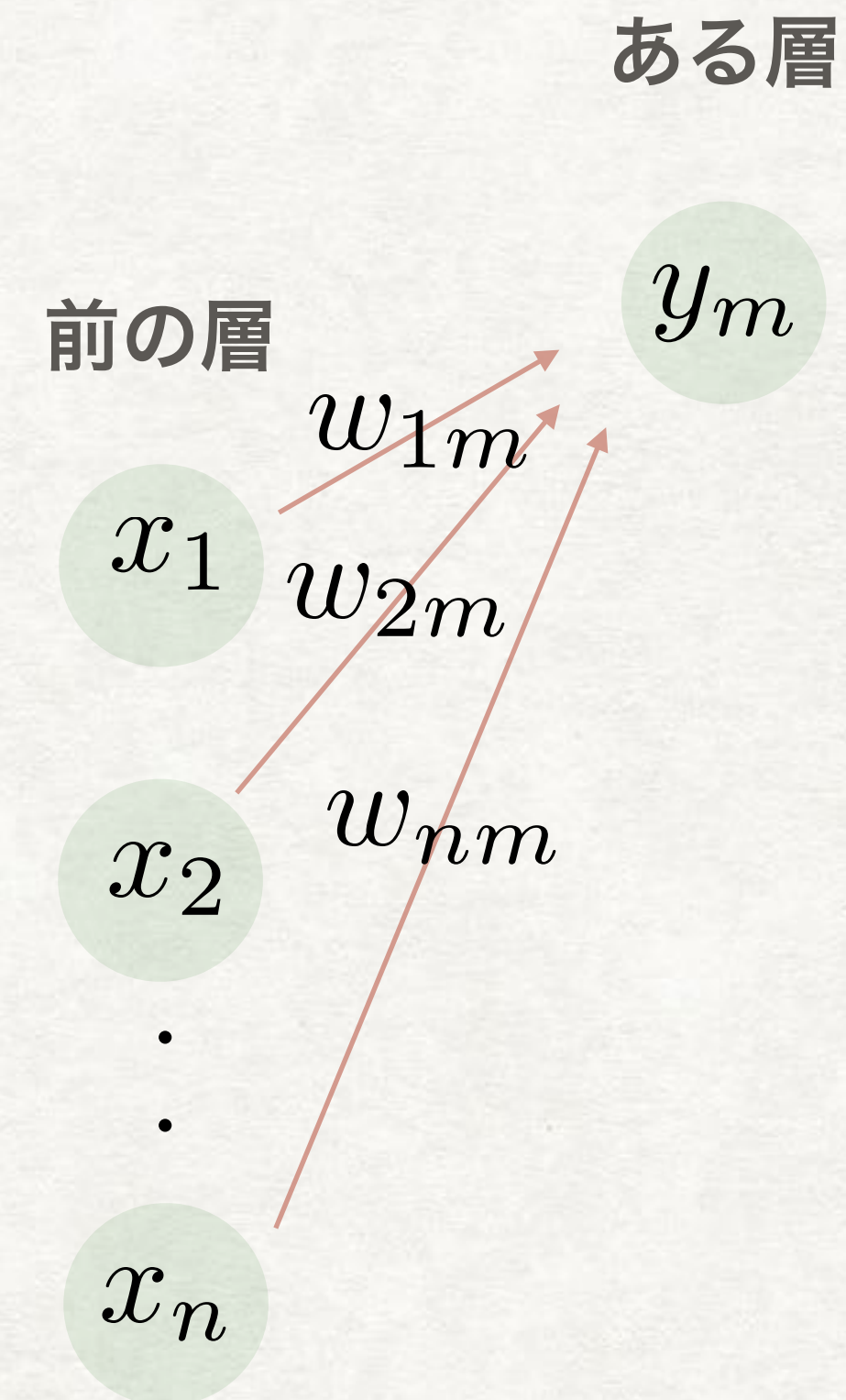
ディープラーニングの動作

1. ラベル付きデータを入力
とし出力値を計算

y_m を活性化関数(非線形関数)で変換

$$y_m = f(y_m)$$

$f(x)$: 活性化関数



0 機械学習・ディープラーニングとは

ディープラーニングの動作

2. ラベルと出力値の誤差を計算

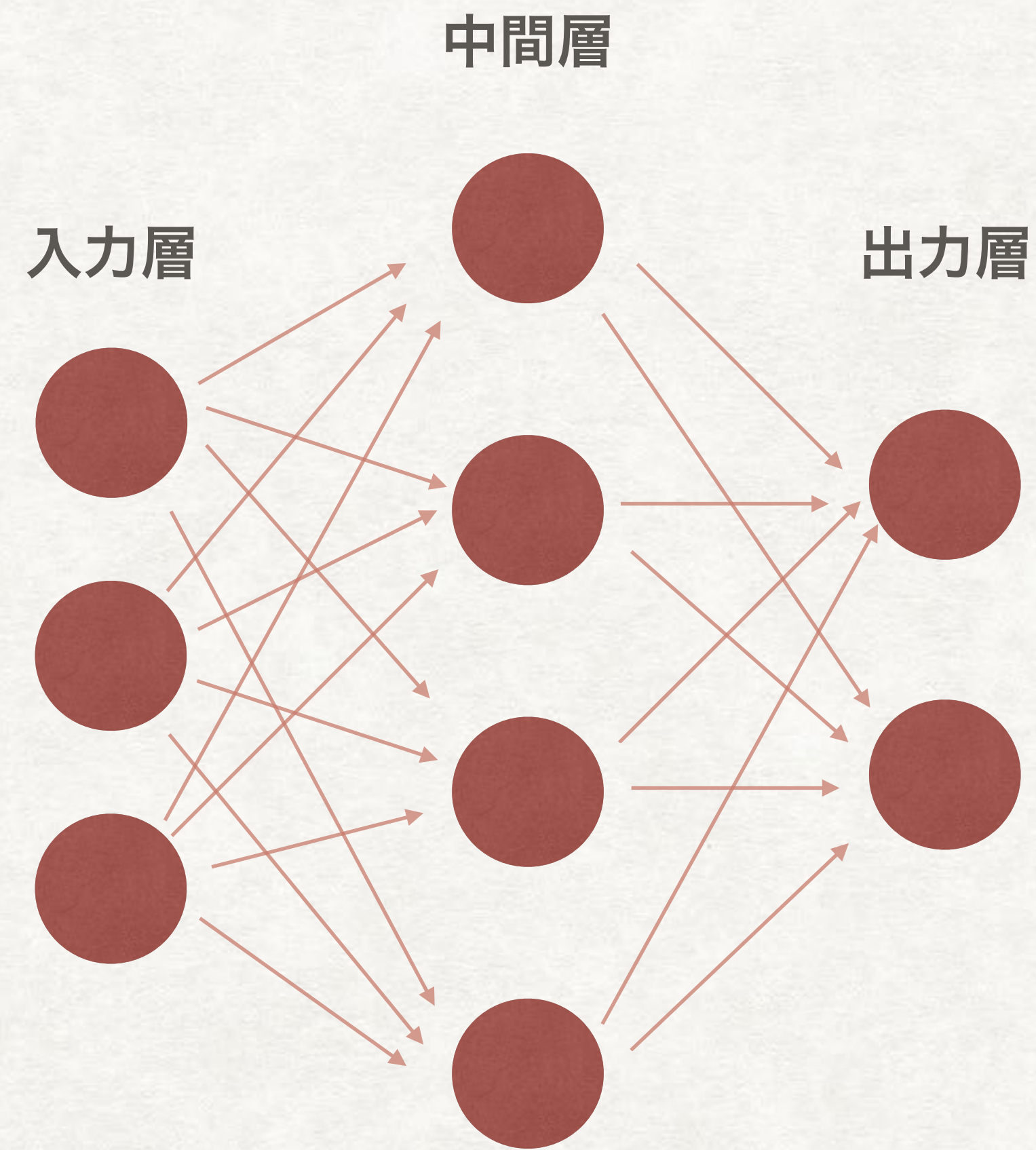
誤差関数

- 二乗誤差

$$E = (y_p - y_t)^2$$

y_p : 出力値

y_t : 答え



0 機械学習・ディープラーニングとは

ディープラーニングの動作

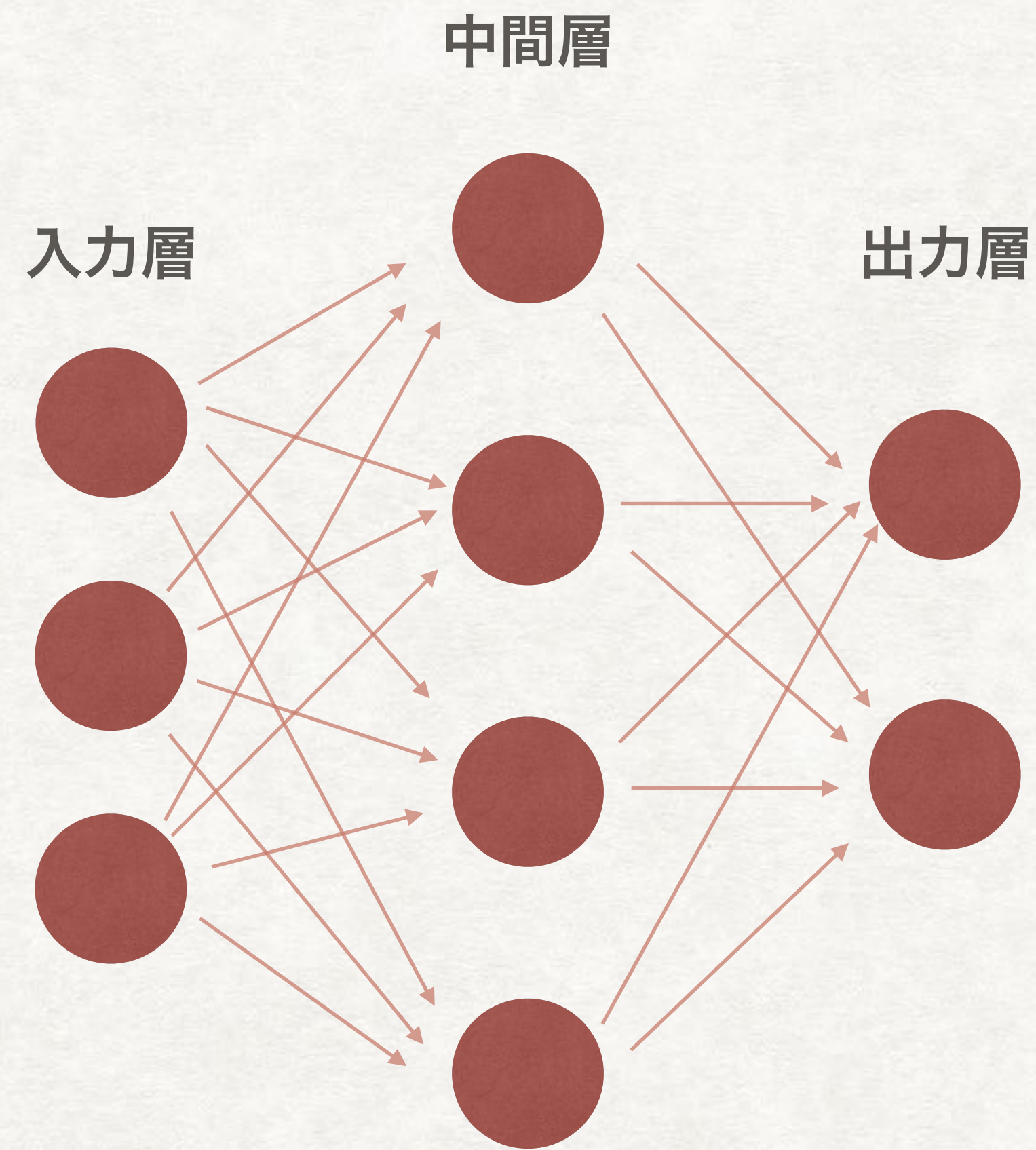
3. 重みを修正

確率的勾配降下法

$$w = w - \eta \frac{\partial E}{\partial w}$$

w : 重み, E : 誤差関数

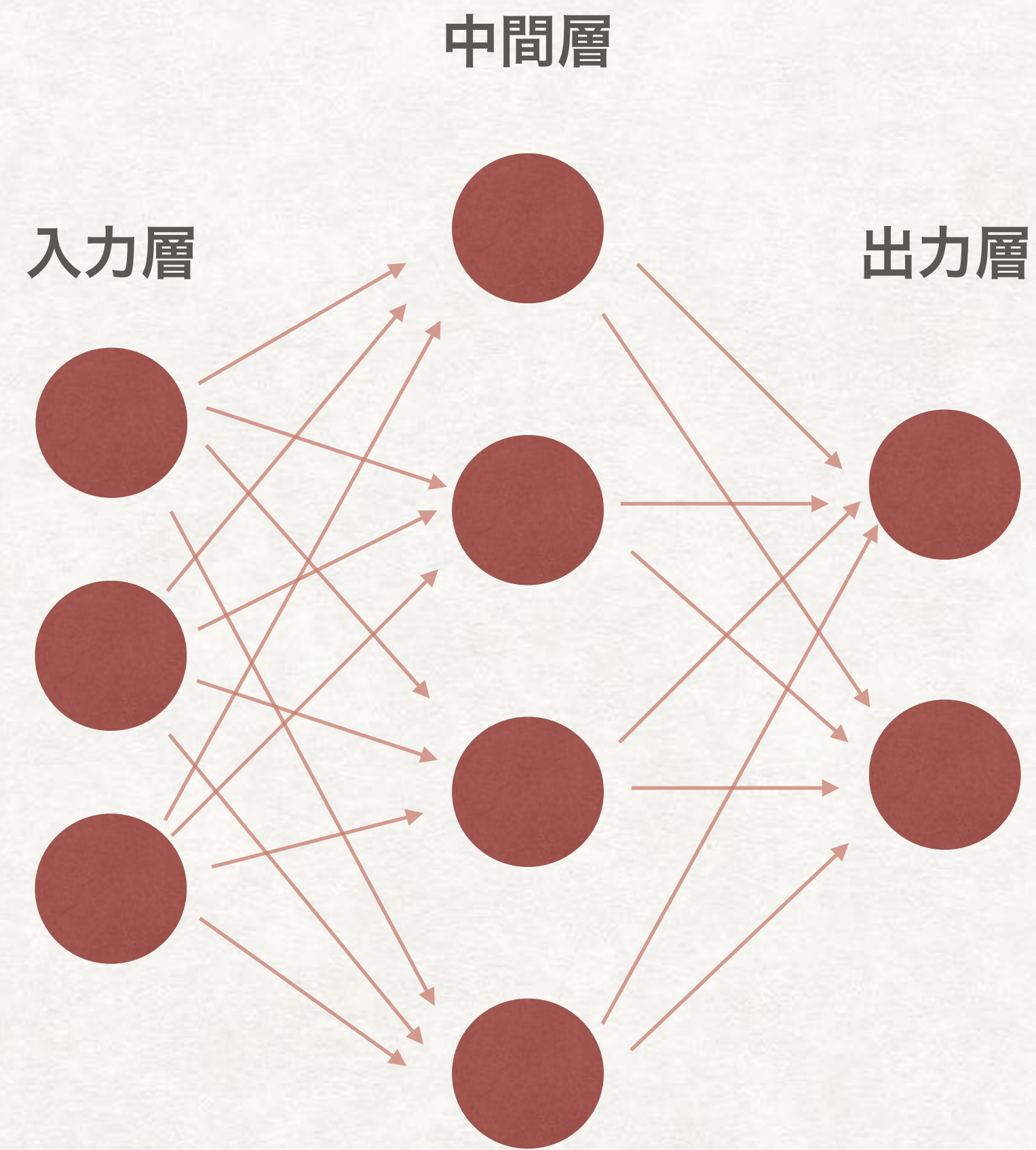
η : 学習率(1回でどの程度更新するか)



0 機械学習・ディープラーニングとは

ディープラーニングの動作

得られた重みを用いて
未知のデータに対する答えを予測



1 はじめに

1-1 競馬とは

競馬

競馬は競技者・関係者が行う「馬を競わせる」
興行に対して、観戦者が勝馬を予想して金を賭ける「賭博」
(Wikipedia)

勝馬投票券(以下馬券)

競馬において「金を賭ける」とは、馬券を購入することを言う。
「単勝式」「複勝式」など複数種類の買い方があり、
予測が的中して入れば配当を得られる。

1 はじめに

1-1 競馬とは

競馬を予測するとは

的中を目的として予測

「最も強い」「最も勝つであろう」
馬を予測

配当を目的として予測

「配当が最も得られるであろう」
馬を予測

1 はじめに

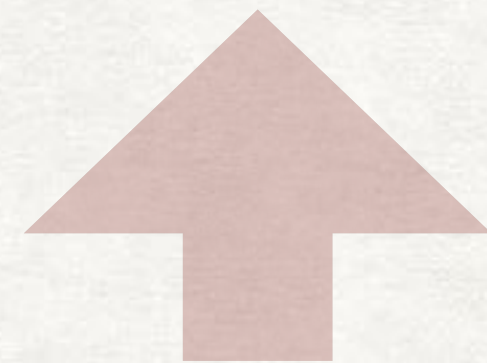
1-1 競馬とは

中央競馬

日本中央競馬会(JRA)が開催

地方競馬

都道府県や地方公共団体等が開催



本演習では中央競馬を扱う

1 はじめに

1-1 競馬とは

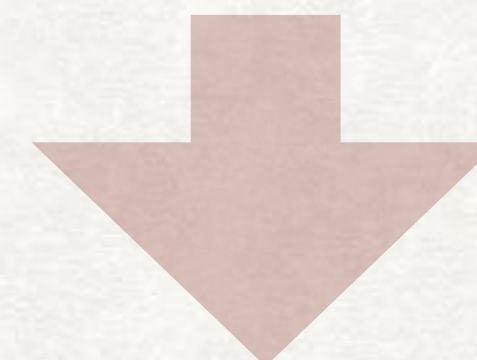
$$\text{配当} = \frac{\text{馬券の売り上げ} \times \text{払戻率}}{\text{的中者の人数}}$$

払戻率
馬券の売り上げに対して
配当に当てられる割合

JRAの払戻率

投票法	単勝	複勝	枠連	馬連	ワイド	馬単	三連複	三連単
払戻率	0.80	0.80	0.775	0.775	0.775	0.75	0.725	0.70

回収率(購入金額に対する配当の比率)が払戻率を超える



予測能力を持つ

1 はじめに

1-2 機械学習を用いた競馬予測の事例

AlphaImpact (<https://alphaimpact.jp/>)

- ・ 的中型人工知能

(http://yoso.netkeiba.com/nar/?pid=yosoka_profile&id=101)

回収率:**84%** 的中率:**85%**

- ・ 回収型人工知能

(http://yoso.netkeiba.com/nar/?pid=yosoka_profile&id=100)

回収率:**81%** 的中率**18%**

回収率,的中率は1月24日現在過去1ヶ月のもの

1 はじめに

1-3 競馬予測アルゴリズムを競う大会

電腦賞(春) (<http://ch.nicovideo.jp/dennoushou>)

指定のレースを競馬予測アルゴリズムを使い予測し回収率を競う

ドワンゴ主催

前回大会優勝の「ニコちゃんAI競馬」は回収率181.9%を記録

1 はじめに

1-4 演習目的・内容

ディープラーニングを用いて競馬予測を行う2種類のモデルを作成

3着内予測モデル

3着内と3着外に分類

3着に入るかどうかという予測

期待値予測モデル

配当の期待値を出力

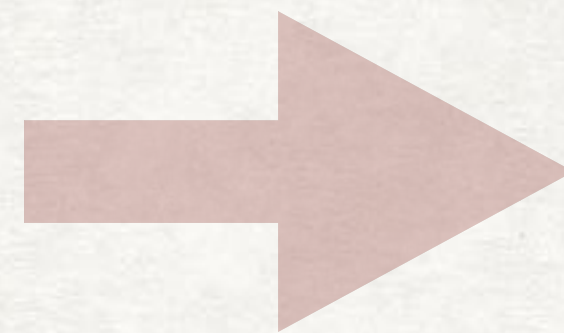
いくら配当が得られるか予測

ラベル付け

着順が3以上 : 0
着順が3以下 : 1

着順が3以上 : 0
着順が3以下 : 複勝の配当/1000

Webページから
情報を取得



1 はじめに

1-4 演習目的・内容

ディープラーニングを用いて競馬予測を行う2種類のモデルを作成

3着内予測モデル

3着内と3着外に分類

3着に入るかどうかという予測

期待値予測モデル

配当の期待値を出力

いくら配当が得られるか予測

取得した
レース情報

芝コース

ダートコース

芝モデル

ダートモデル

芝モデル

ダートモデル

2 演習方法

開発環境

言語	Python3.5
ライブラリ	Keras
	Panda
	Numpy
	Beautiful Soup
	Selenium
	sckit-learn
	fastText

2 演習方法

実行環境

	MacBook Pro (Early 2013)	Mac mini (Mid 2011)
CPU	Intel Core i5 2.6GHz	Intel Core i7 2.7GHz
メモリ	8GB	8GB

2 演習方法

競馬予測プログラムの作成する手順

Webページから情報を取得

データの前処理・整形

DNNモデルの構築



2 演習方法

競馬予測プログラムの作成する手順

Webページから情報を取得



2 演習方法

2-1 WEBページから情報を取得

競馬情報サイト 「netkeiba」
(<http://www.netkeiba.com/>)

情報をCSV形式で出力

Webスクレイピングを行い
情報を取得するプログラムを作成

- beautiful soup
- selenium
を使用

2 演習方法

2-1 WEBページから情報を取得 (1)取得するデータの内容

レースデータ

芝ダート,右左周り,距離,
競馬場,何日目,何レース,
馬場状態,1着賞金

血統データ

父,母,父父,父母,母父,...

馬のデータ

枠番,着順,オッズ,
牝馬牡馬,馬齢,斤量,
騎手,体重,体重前走比,
所属,脚質,休み明け,調教コース,
調教脚色,調教評価,
1~5走前(着順,着差,上がりタイム,
芝ダート,距離,グレード)

2 演習方法

2-1 WEBページから情報を取得

データ数 253785件
2010年1月から2017年9月

レースデータ+馬のデータ

ダ,右,1000,1,1,2,良,500,4.0,1,牡,3,56.0,吉田隼,0,480,-2.0,東,3,5.0,1.0,35.6,稍,
ダ,1000,未勝利,7.0,2.3,36.8,不,ダ,1000,未勝利,10.0,1.7,36.2,良,ダ,1000,未勝
利,14.0,2.3,37.9,良,ダ,1150,未勝利,10.0,1.8,38.2,良,ダ,1200,未勝利,0,札ダ,一杯

血統データ

キタサンブラック ブラックタイド シュガーハート サンデーサイレンス ウインドイン
ハーヘア サクラバクシンオー オトメゴコロ Halo WishingWell Also Burghclere
サクラユタカオー サクラハゴロモ ジャッジアングルーチ ライズリー

2 演習方法

競馬予測プログラムの作成する手順

Webページから情報を取得

データの前処理・整形

芝,右,1800,4,1,2,良,
,500,5,1,牡,3,56,武豊,
0,510,-5,東,2,...

2 演習方法

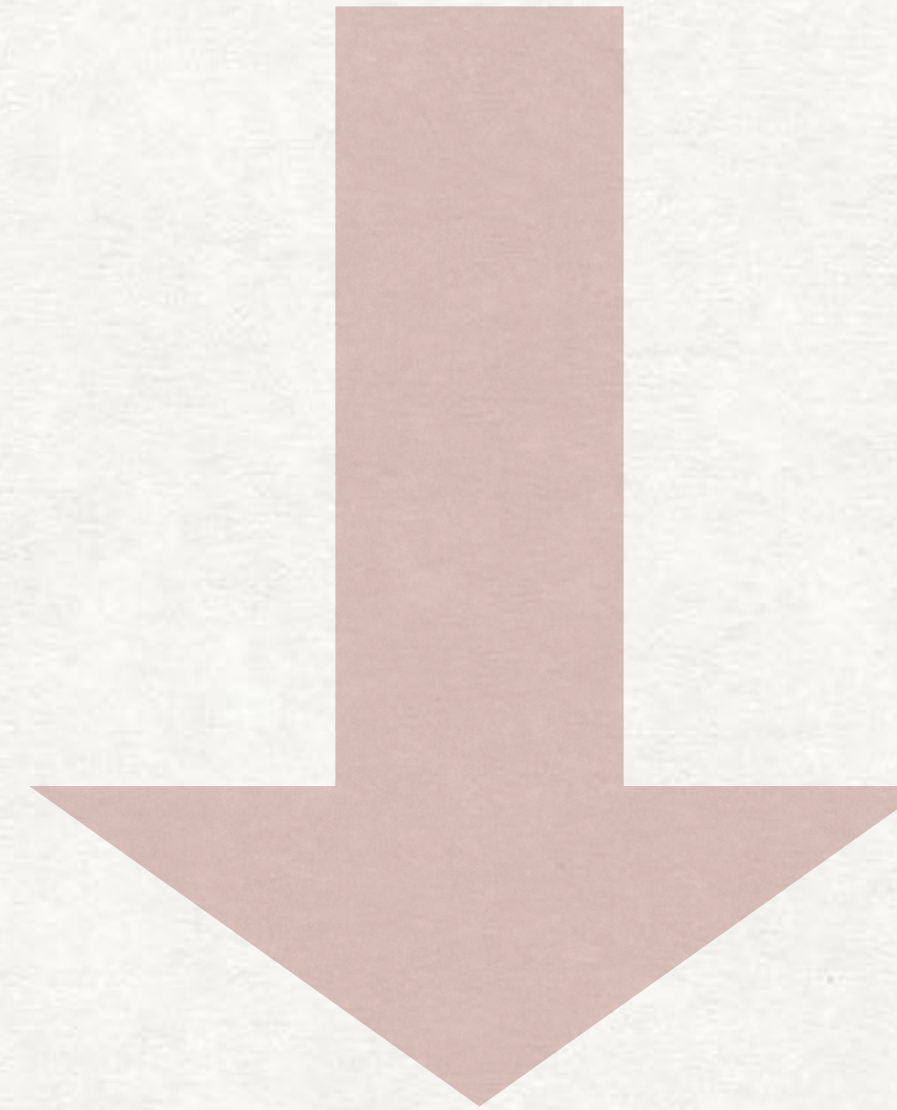
2-2 データの前処理・整形

文字列データの数値化

数値データ標準化

検証・学習データの分離

3着外データ数の調整



2 演習方法

2-2 データの前処理・整形 (1)文字列データの数値化

キタサンブラック	ブラックタイド	サンデーサイレンス	Halo
		ウインドインハーヘア	Wishing Well
	シュガーハート	サクラバクシンオー	Also
		オトメゴコロ	Burghclere
			サクラユタカオー
			サクラハゴロモ
			ジャッジアンジュルーチ
			テイズリー

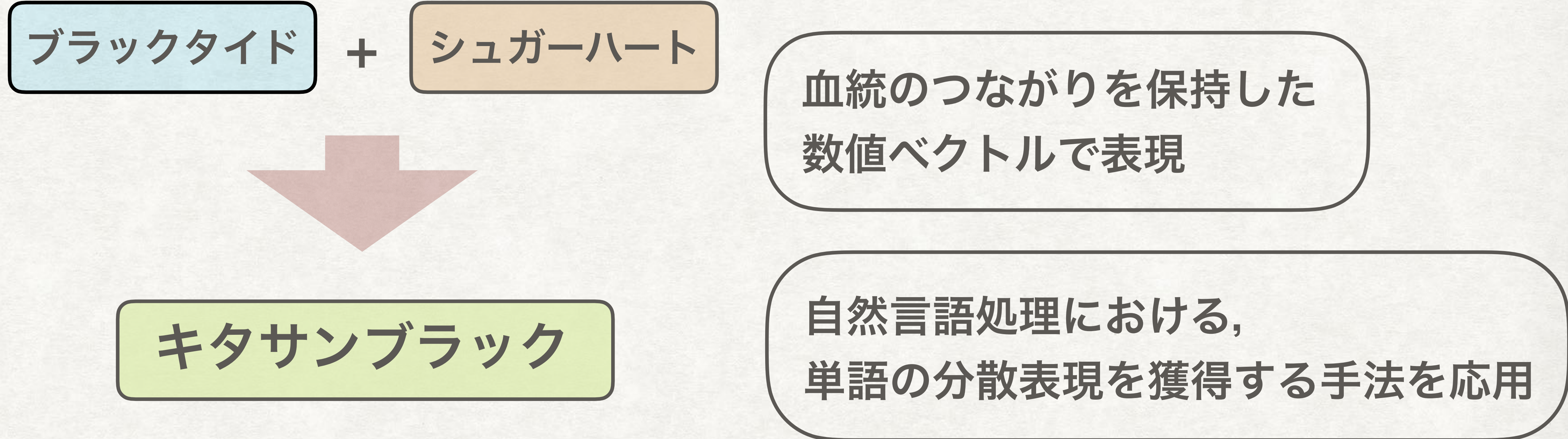
血統図を100次元の数値ベクトルに変換する

(参考)fastTextを用いた競馬血統のベクトル化

<https://qiita.com/KTaskn/items/e76551191214593c278e> (参照日時:2018/1/28)

2 演習方法

2-2 データの前処理・整形 (1) 文字列データの数値化



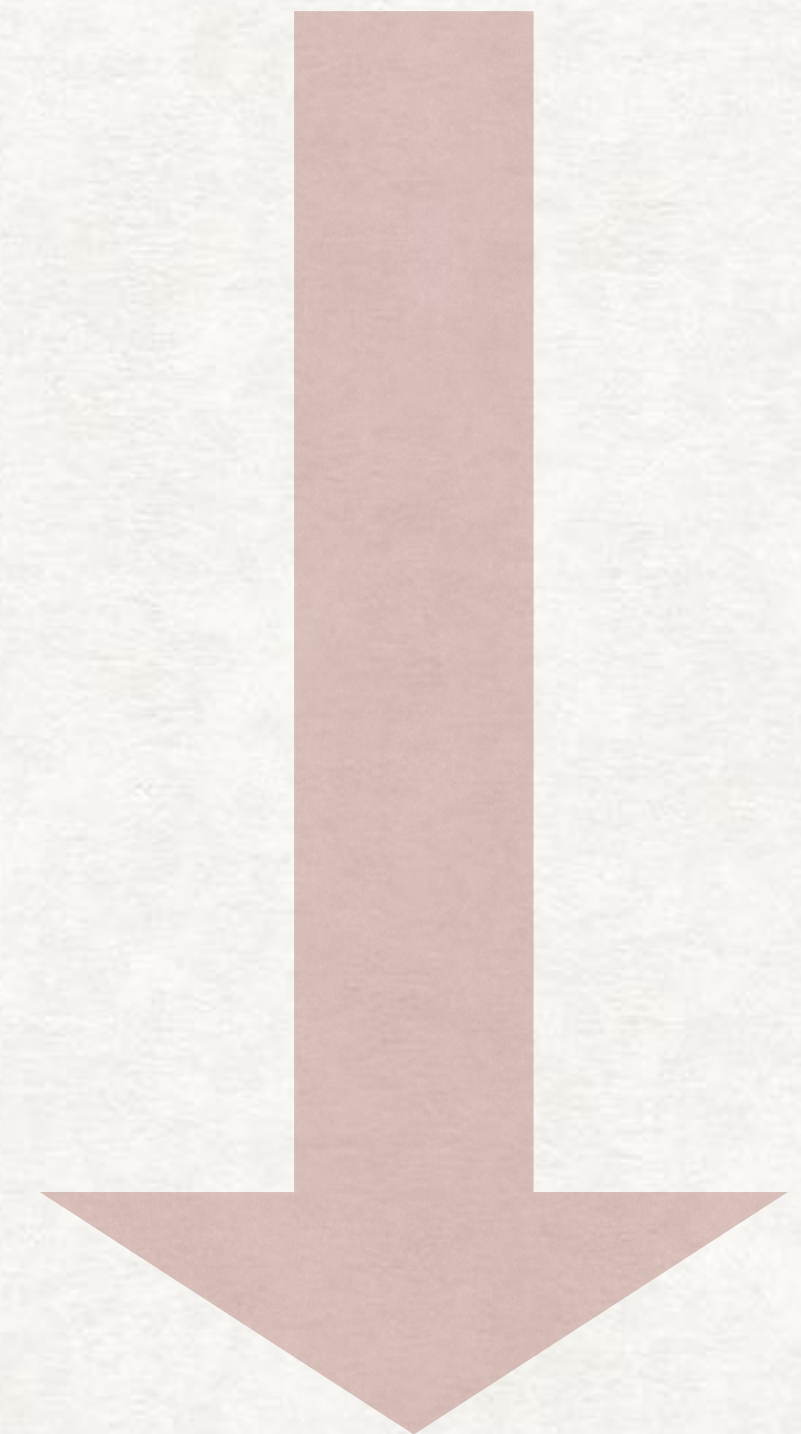
2 演習方法

2-2 データの前処理・整形 (1)文字列データの数値化

血統図を1行のCSV形式(空白区切)で取得
[子 父 母 父父 父母 母父 母母 父父父 ...]

子を中心に左右に広がる様並べ替え
[... 父母 父父 父 子 母 母父 母母 ...]

fastText(自然言語処理ツール)を用いて100次元のベクトル化



2 演習方法

2-1 データの前処理・整形 (1)文字列データの数値化

文字列データ,
数値の大きさに意味のないデータを
one-hot値に変換

sckit-learnの
「get_dummies」を使用

騎手			
武豊	騎手_武豊	騎手_横山典弘	騎手_福永祐一
	1	0	0
横山典弘	0	1	0
福永祐一	0	0	1

2 演習方法

2-2 データの前処理・整形 (2)数値データの標準化

sckit-learnの
「scale」を使用

数値の大きいデータがモデルに
与える影響が大きい

各データ規格を揃える必要がある

標準化

それぞれの特徴量が平均0分散1に分布するように変換

$$x = \frac{x - \sigma}{\mu}$$

σ :平均 μ :標準偏差

例)距離1000~3600m ,馬齢 2~10歳前後

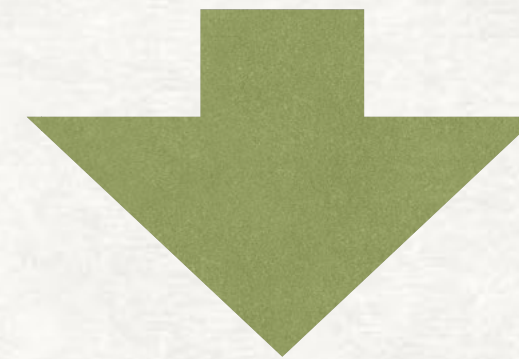
2 演習方法

2-3 データの前処理・整形 (3) 3着外データ数の調整

データ数の差が大きいと
うまく学習できない

3着内データ

3着外データ



3着内データ

3着外データ

2 演習方法

2-3 データの前処理・整形



2 演習方法

競馬予測プログラムの作成する手順

Webページから情報を取得

データの前処理・整形

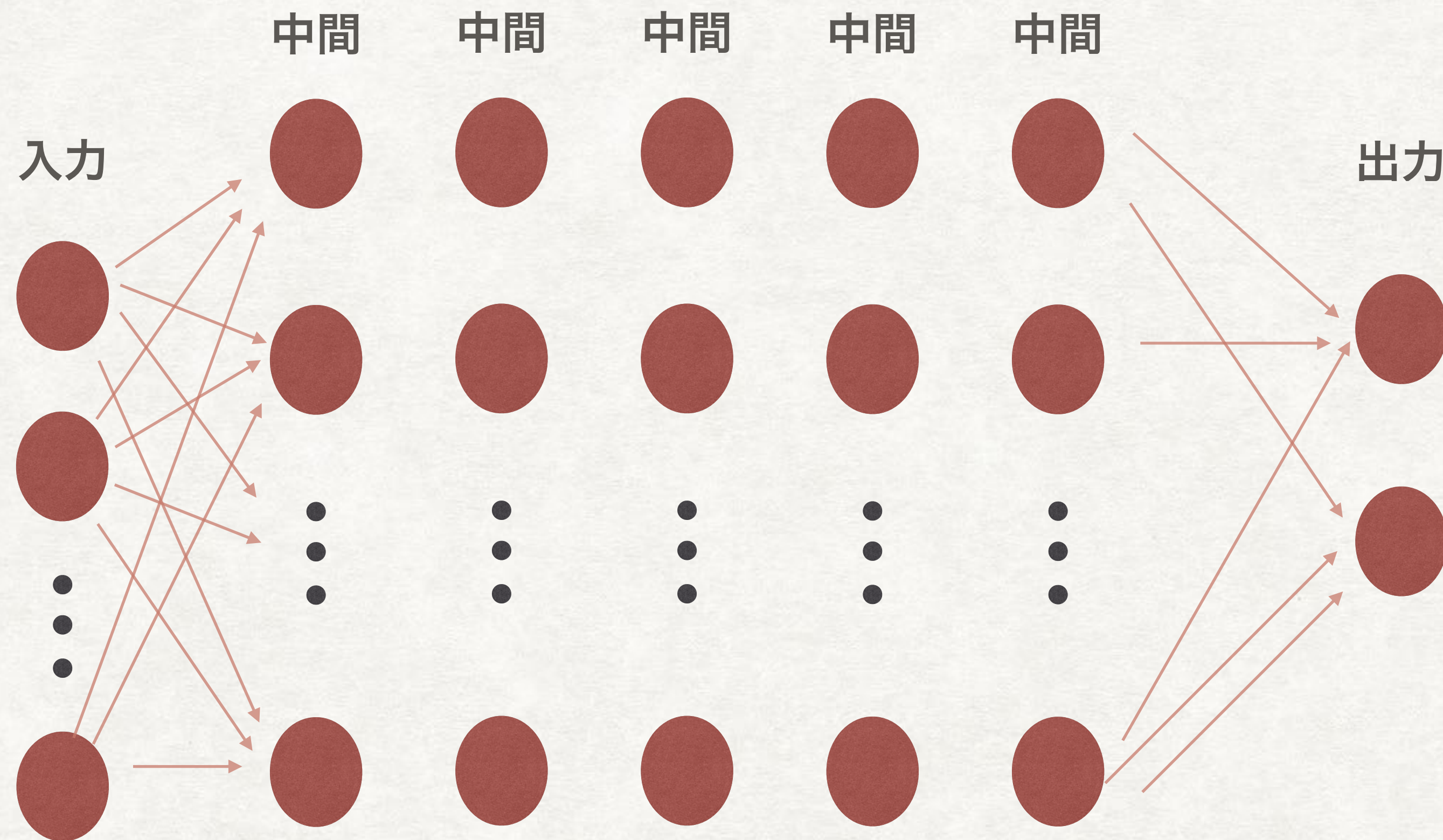
DNNモデルの構築

芝,右,1800,4,1,2,良,
,500,5,1,牡,3,56,武豊,
0,510,-5,東,2,...

0.0, 0.1700, 0.0,
0.209999,0.0, 0.0,
0.0, 0.1400, 0.0,
0.0, 0.0,0,0,...

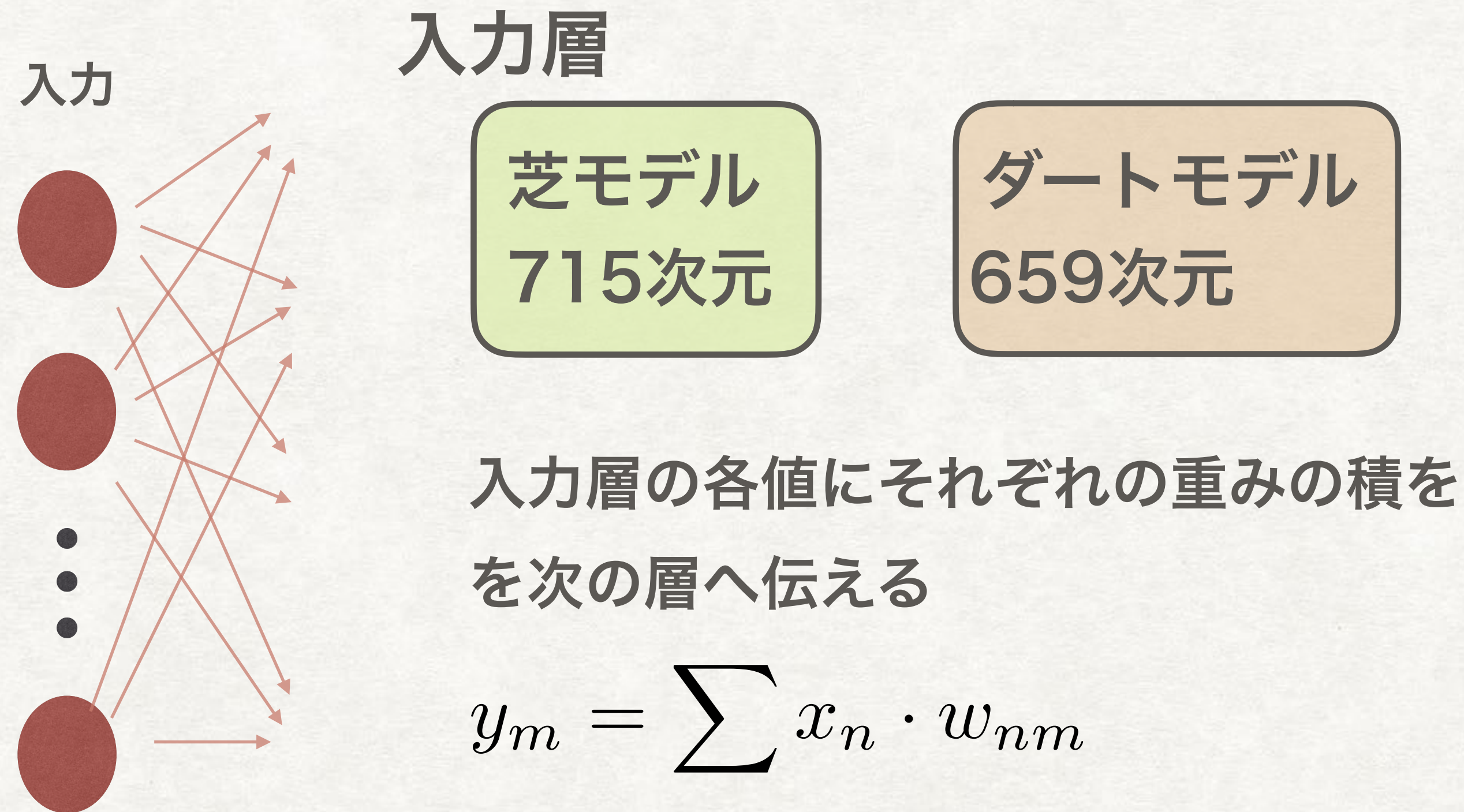
2 演習方法

2-4 DNNモデルの構築



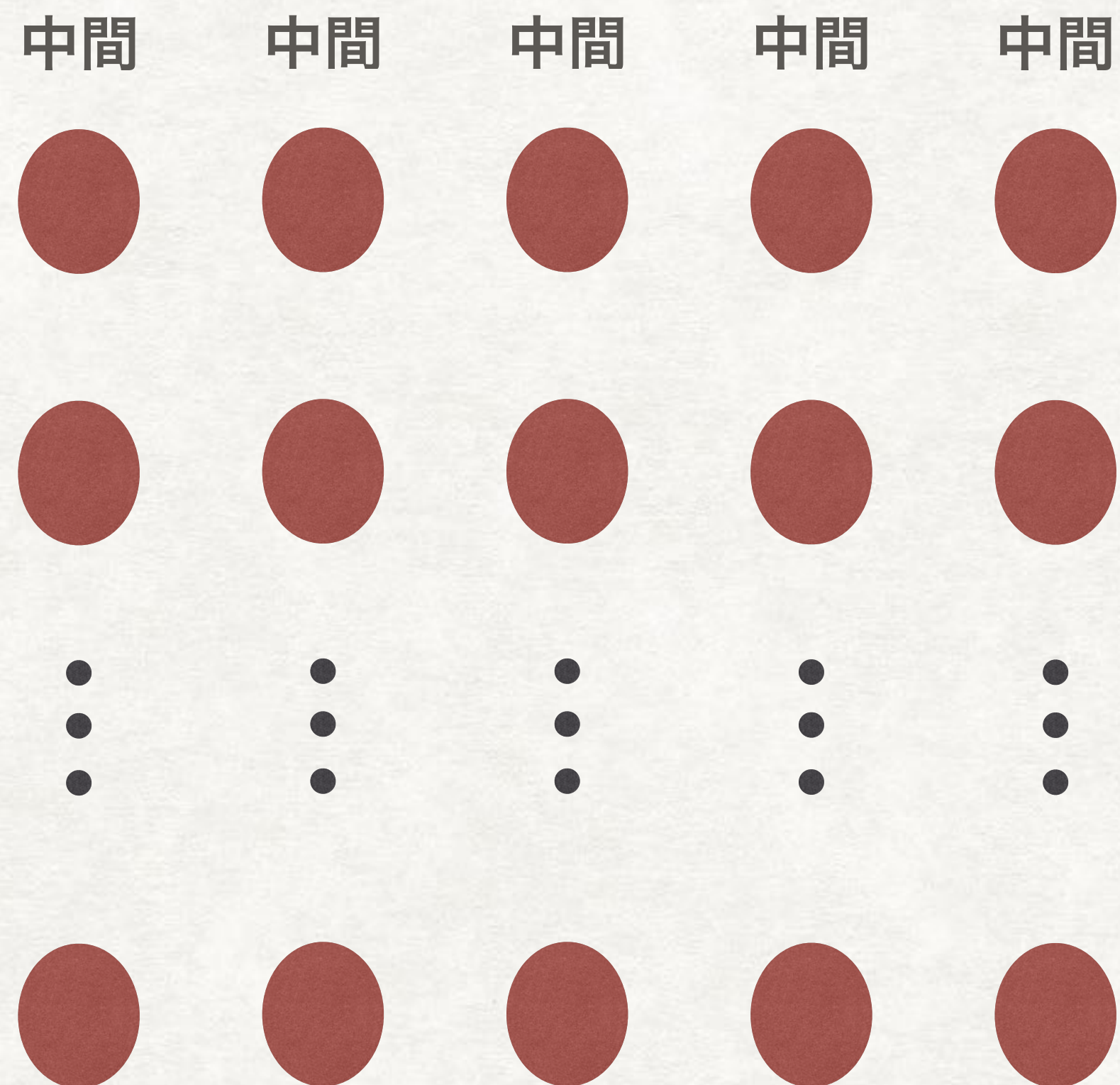
2 演習方法

2-4 DNNモデルの構築 (1) 入力層



2 演習方法

2-4 DNNモデルの構築 (2) 中間層



中間層

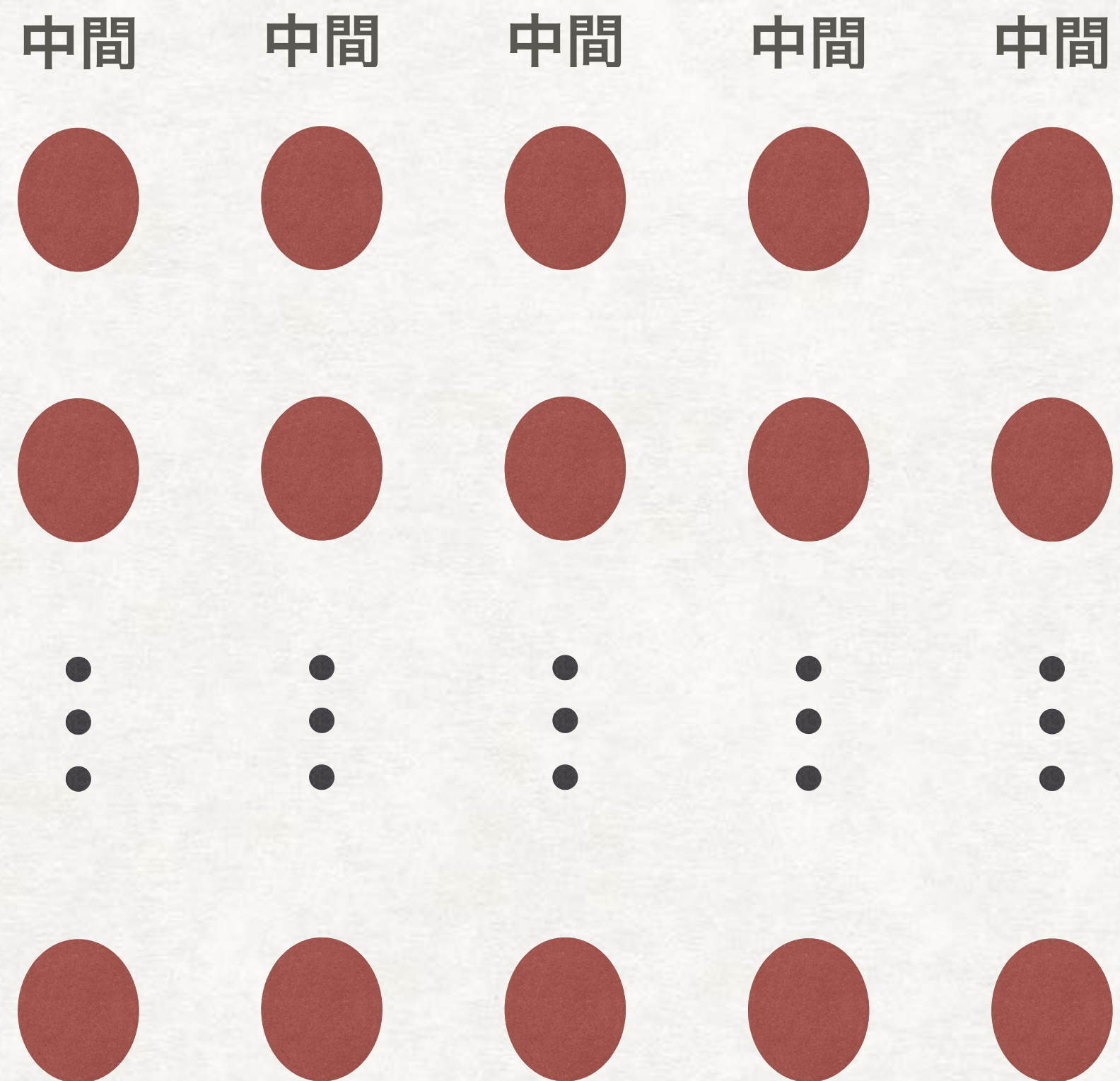
800次元 × 5層

BatchNormaliation

各中間層の値が
平均0分散1に分布するように変換

2 演習方法

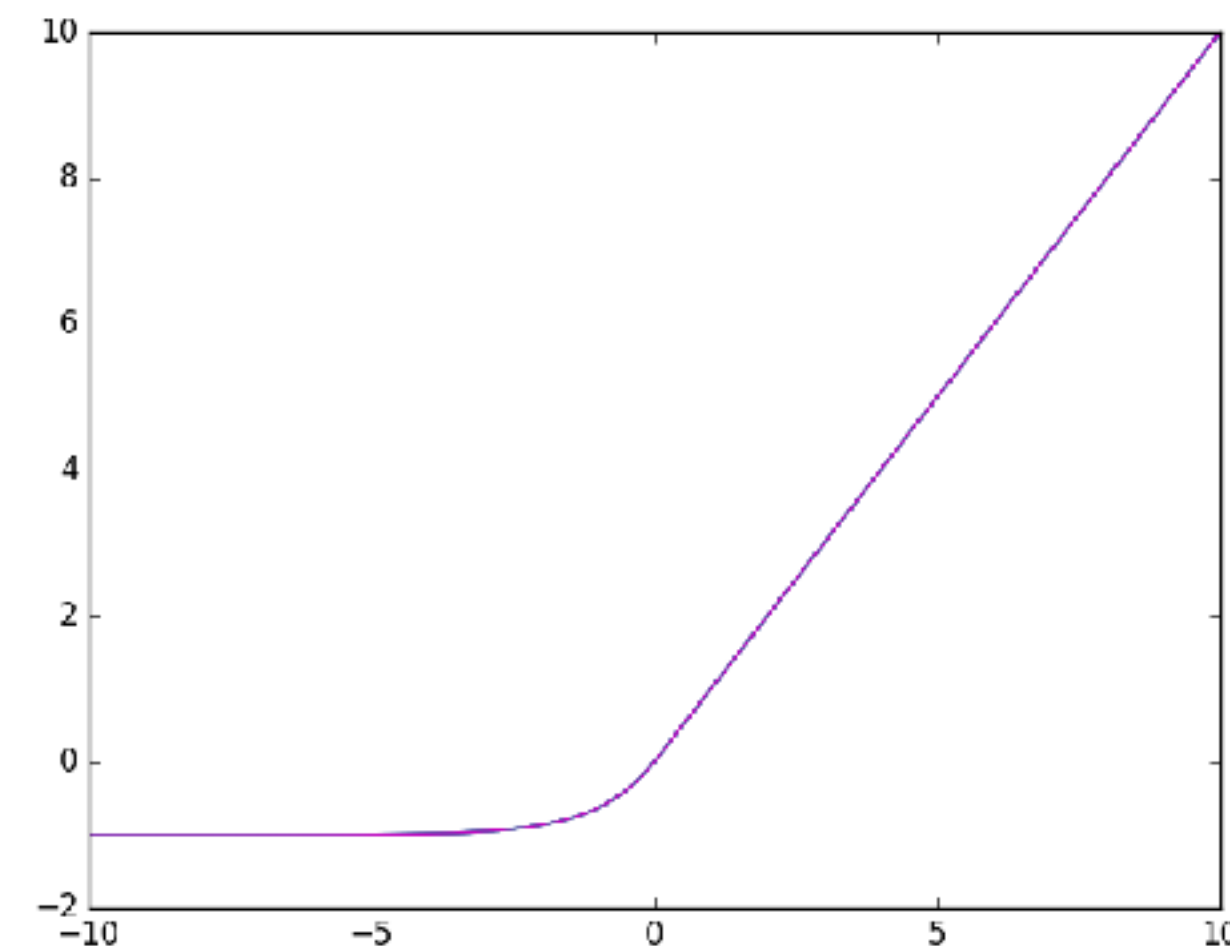
2-4 DNNモデルの構築 (2) 中間層



活性化関数「elu」を用いて変換

$$x = x(x \geq 0)$$

$$x = e^x - 1(x < 0)$$



2 演習方法

2-4 DNNモデルの構築 (3)出力層

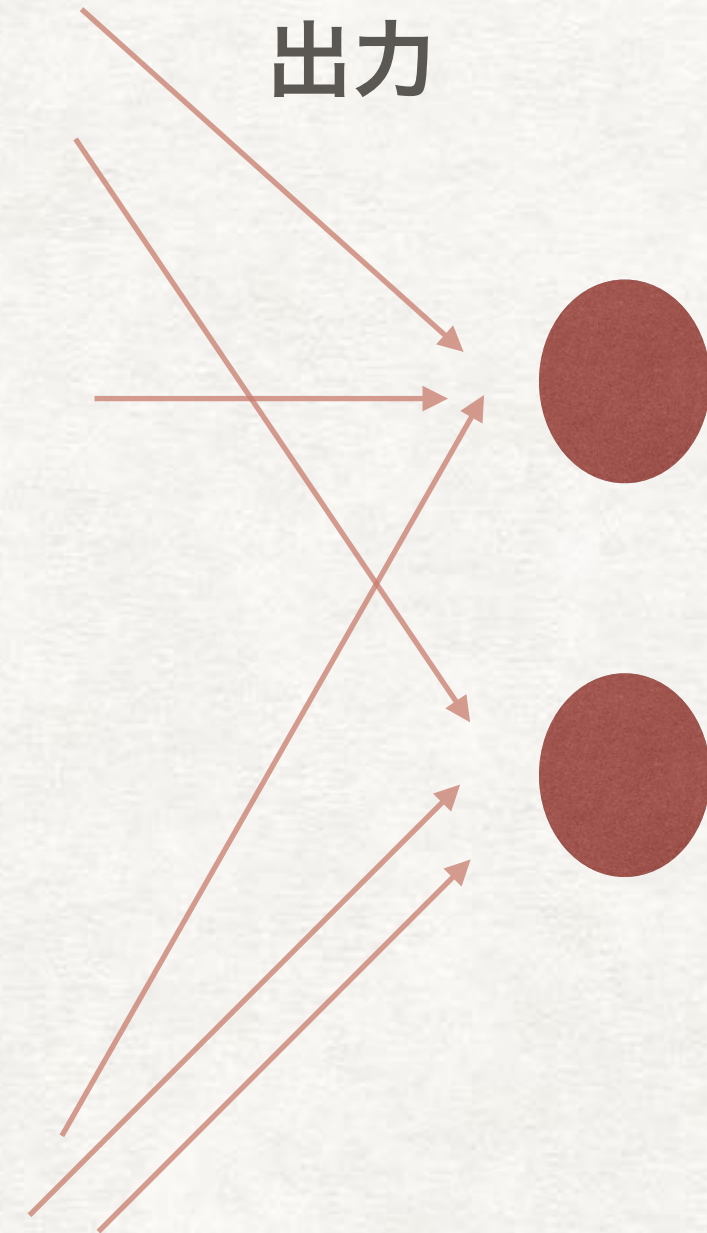
3着内予測モデル

出力層

2次元 [3着外,3着内]

出力の値を「softmax」関数で変換

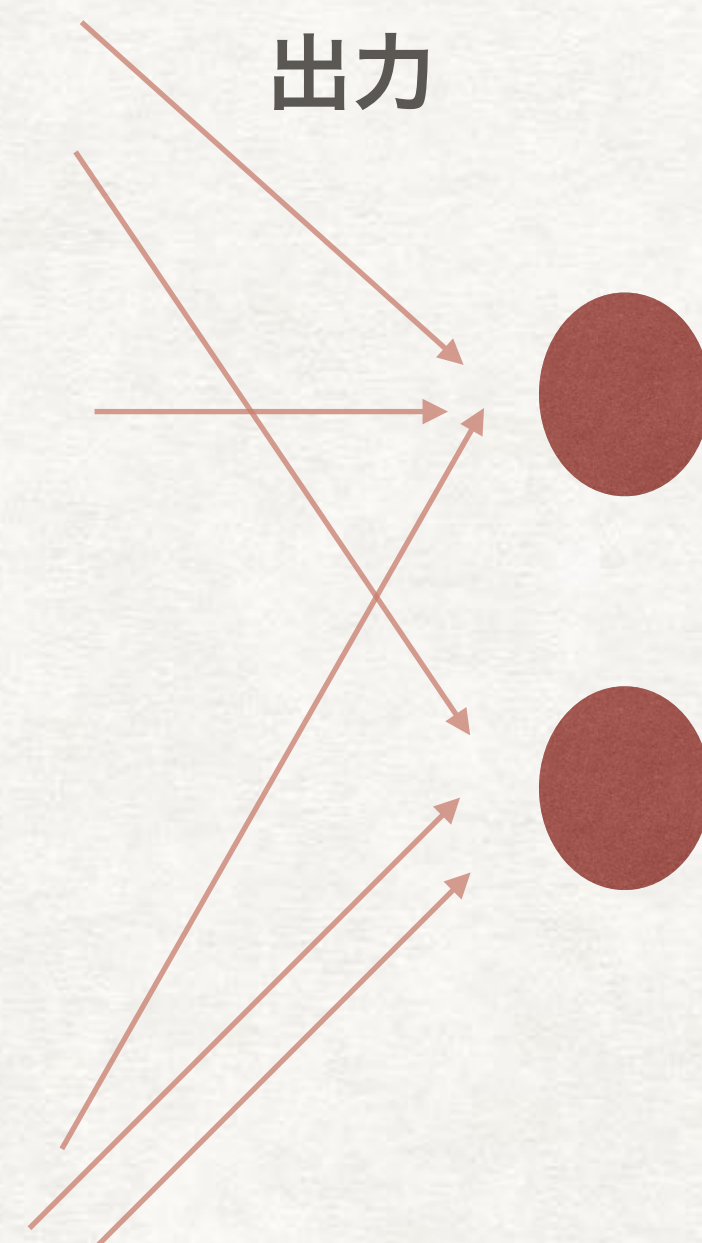
$$y = \frac{e^{y_i}}{\sum e^y}$$



2 演習方法

2-4 DNNモデルの構築 (3)出力層

3着内予測モデル



出力層

[3着外,3着内]

例) [0.01,0.99]

この場合, 3着内であると予測

2 演習方法

2-4 DNNモデルの構築 (3)出力層

配当期待値予測モデル

出力層

1次元 [配当の期待値/1000]

1000倍することで期待値を得る

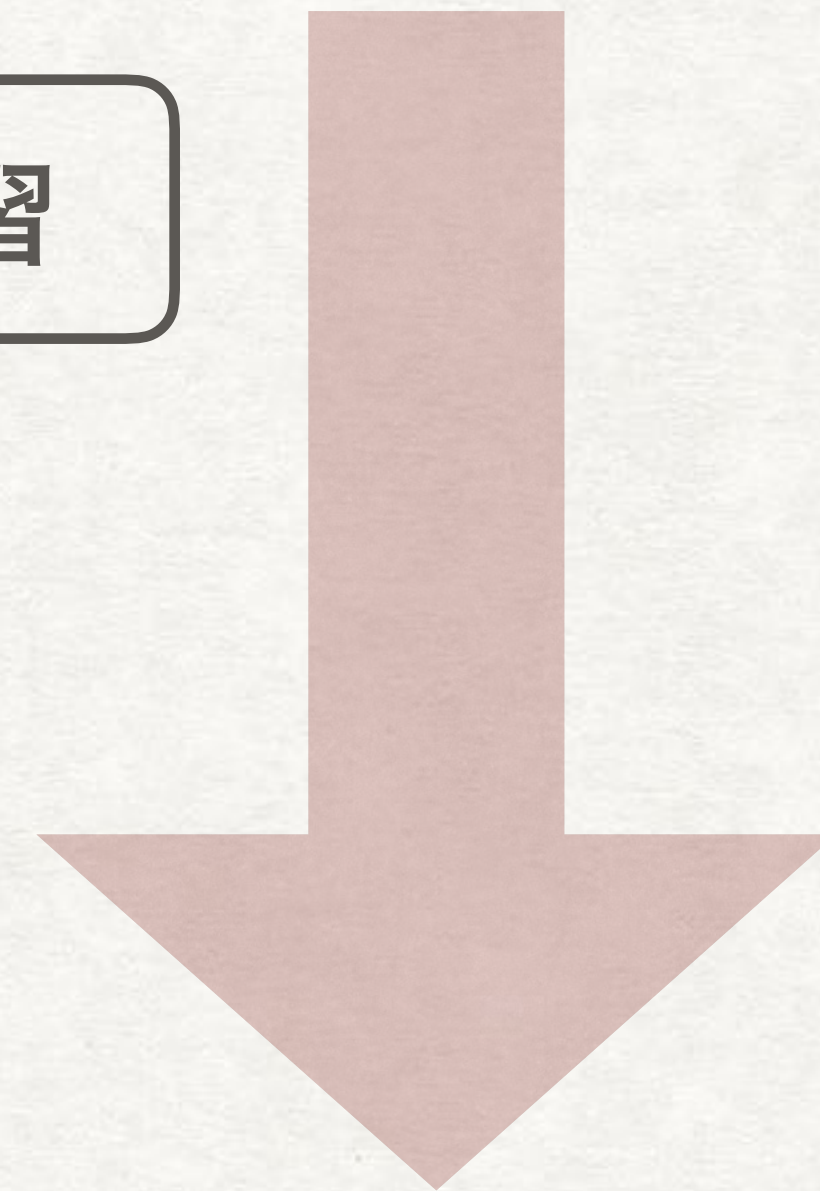
3 結果考察

実験方法

2010年1月から2017年9月までのデータを用いて学習

2017年10月から2017年1月までのレースを予測

予測に基づいて「複勝式(3着以内であれば的中)」を
購入するとする



的中率

回収率

複勝の払戻率

80%

3 結果考察

3着以内予測モデル

出力値が最も大きい馬を
全レース(865レース)購入

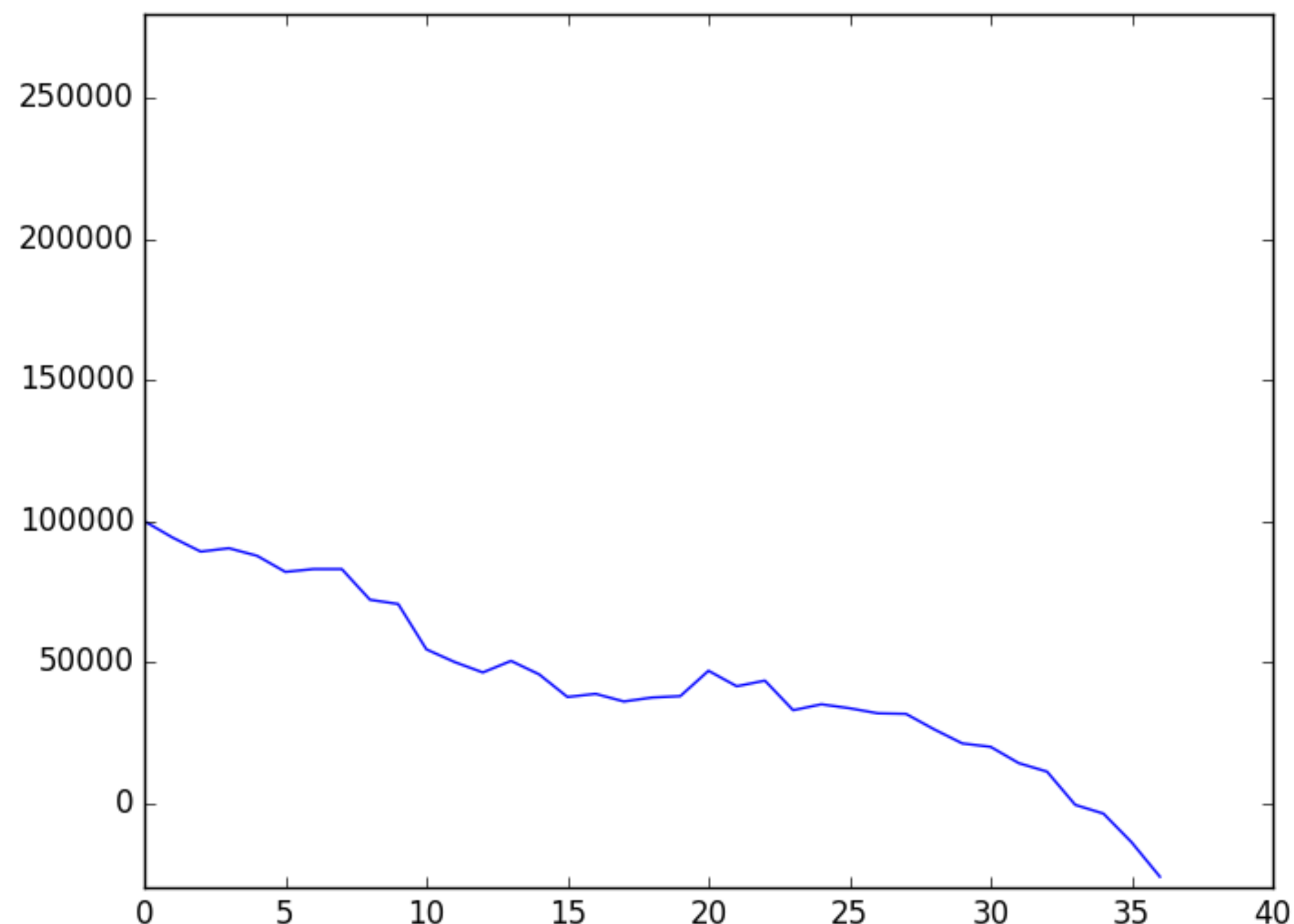
回収率

85%

的中率

49%

初期資金10万円
1レース1000円ずつ購入



3 結果考察

3着以内予測モデル

出力最大と2番目の馬の
値の差が0.1以上の
レースのみ(340レース)購入

回収率

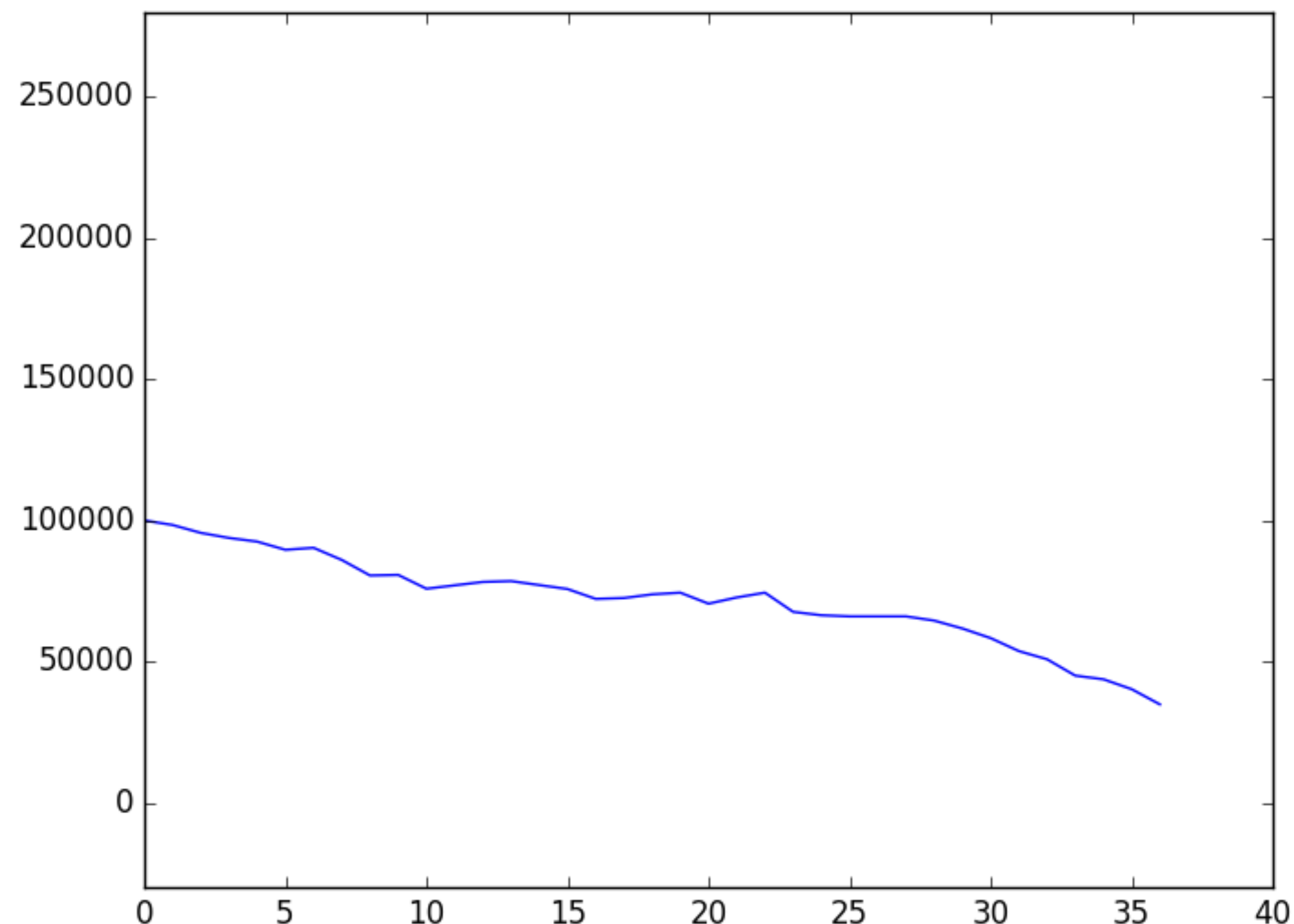
80%

的中率

55%

初期資金10万円

1レース1000円ずつ購入



3 結果考察

配当期待値予測モデル

出力値が最も大きい馬を
全レース(843レース)購入

回収率

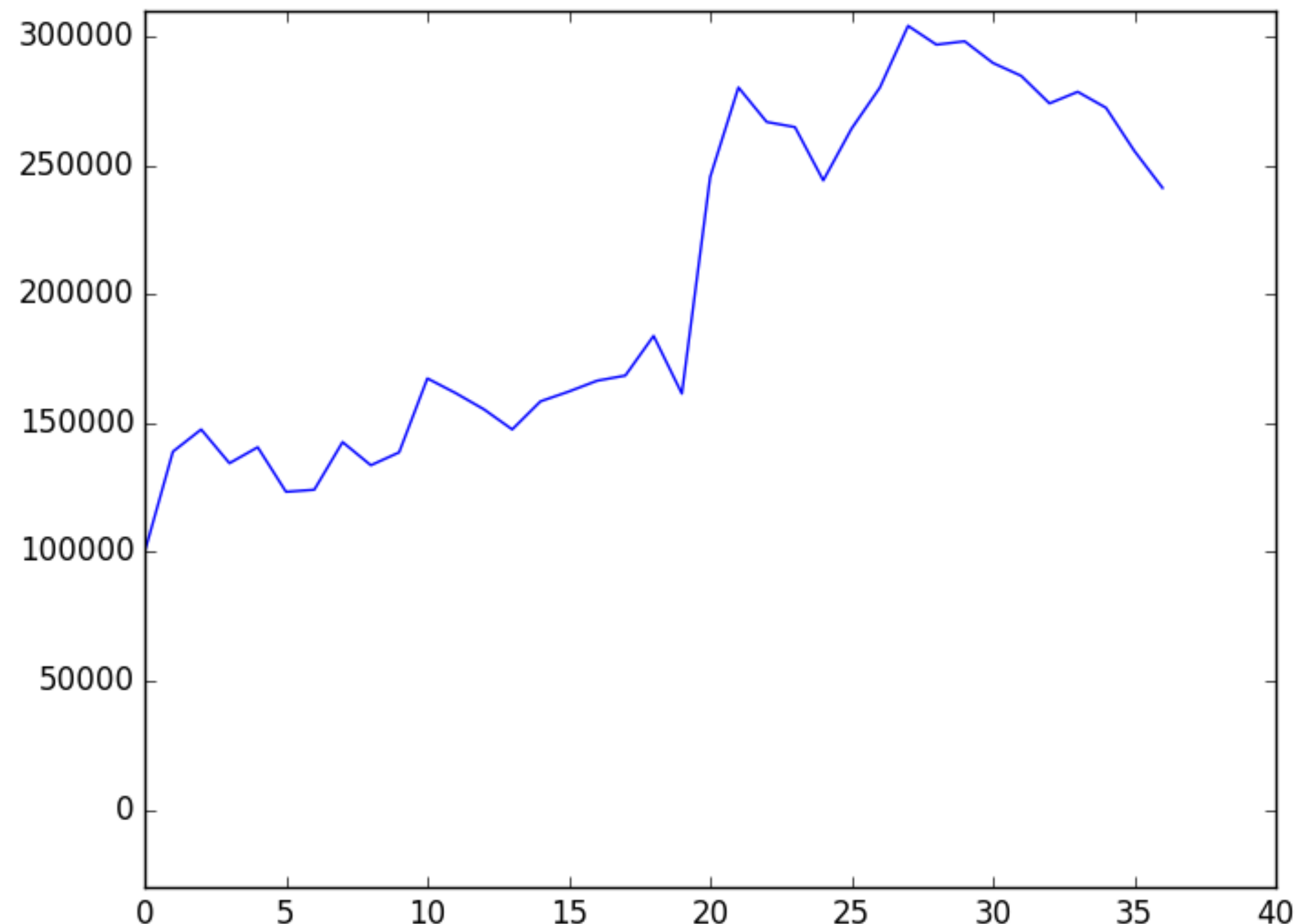
116%

的中率

26%

初期資金10万円

1レース1000円ずつ購入



3 結果考察

配当期待値予測モデル

出力最大と2番目の馬の
値の差が10以上の
レースのみ(329レース)購入

回収率

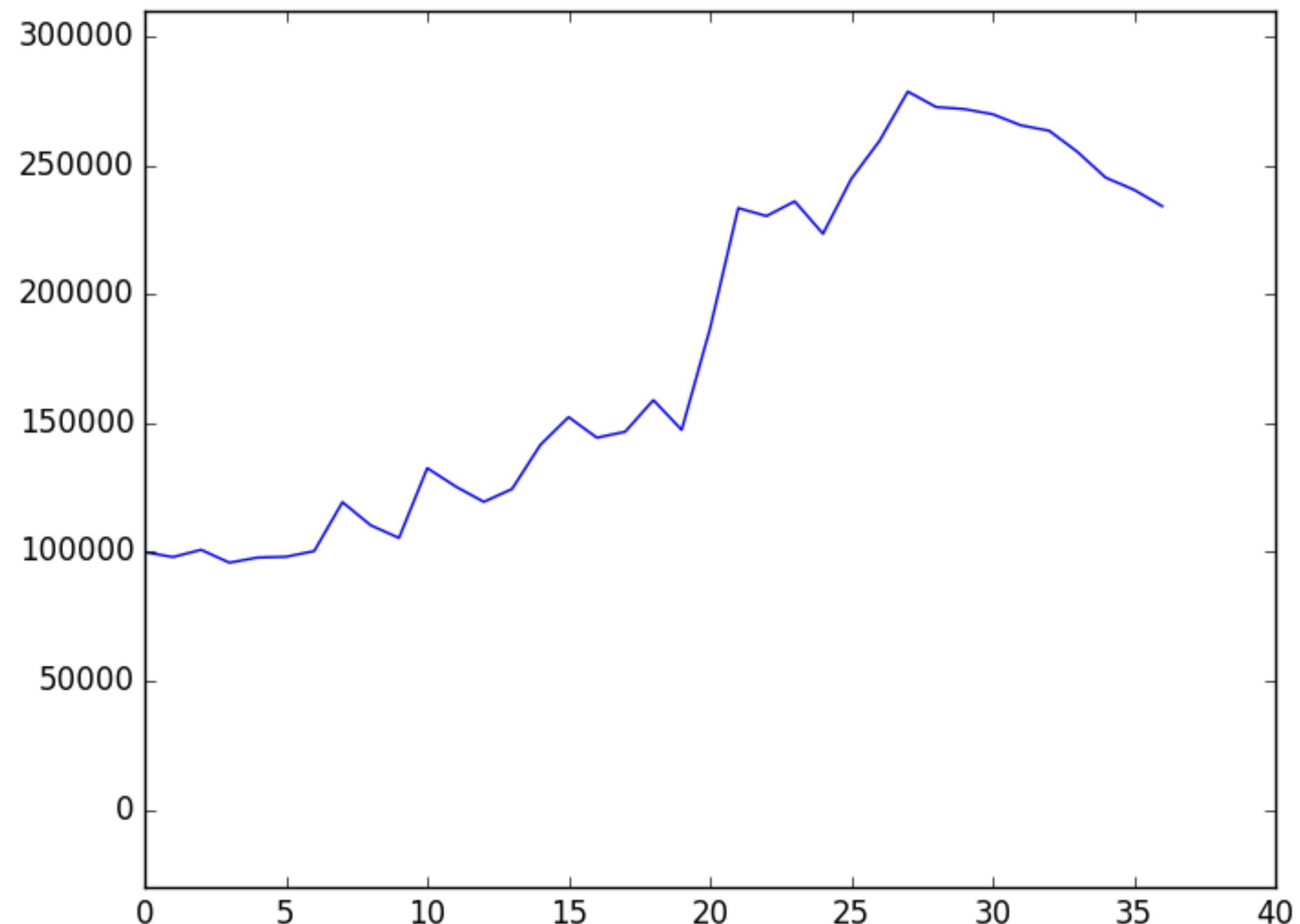
140%

的中率

23%

初期資金10万円

1レース1000円ずつ購入



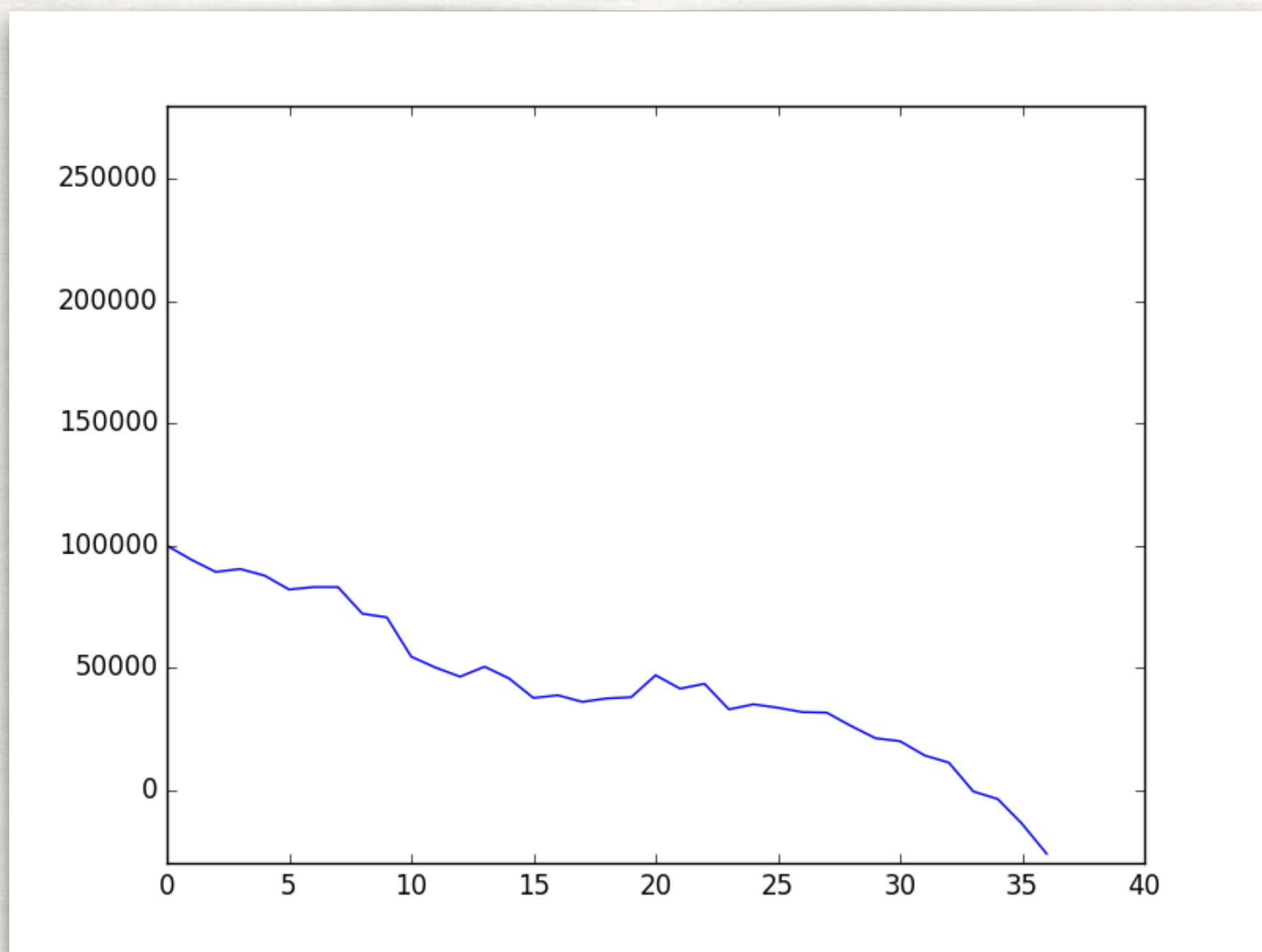
3 結果考察

今後の課題

- より多くの特徴量を用いた予測
- 2つのモデルの出力を複合的に利用した予測
- 「調教脚色,調教評価」の測定した人の主観のデータ
から客観データへの置き換え

3 結果考察

3着内予測モデル

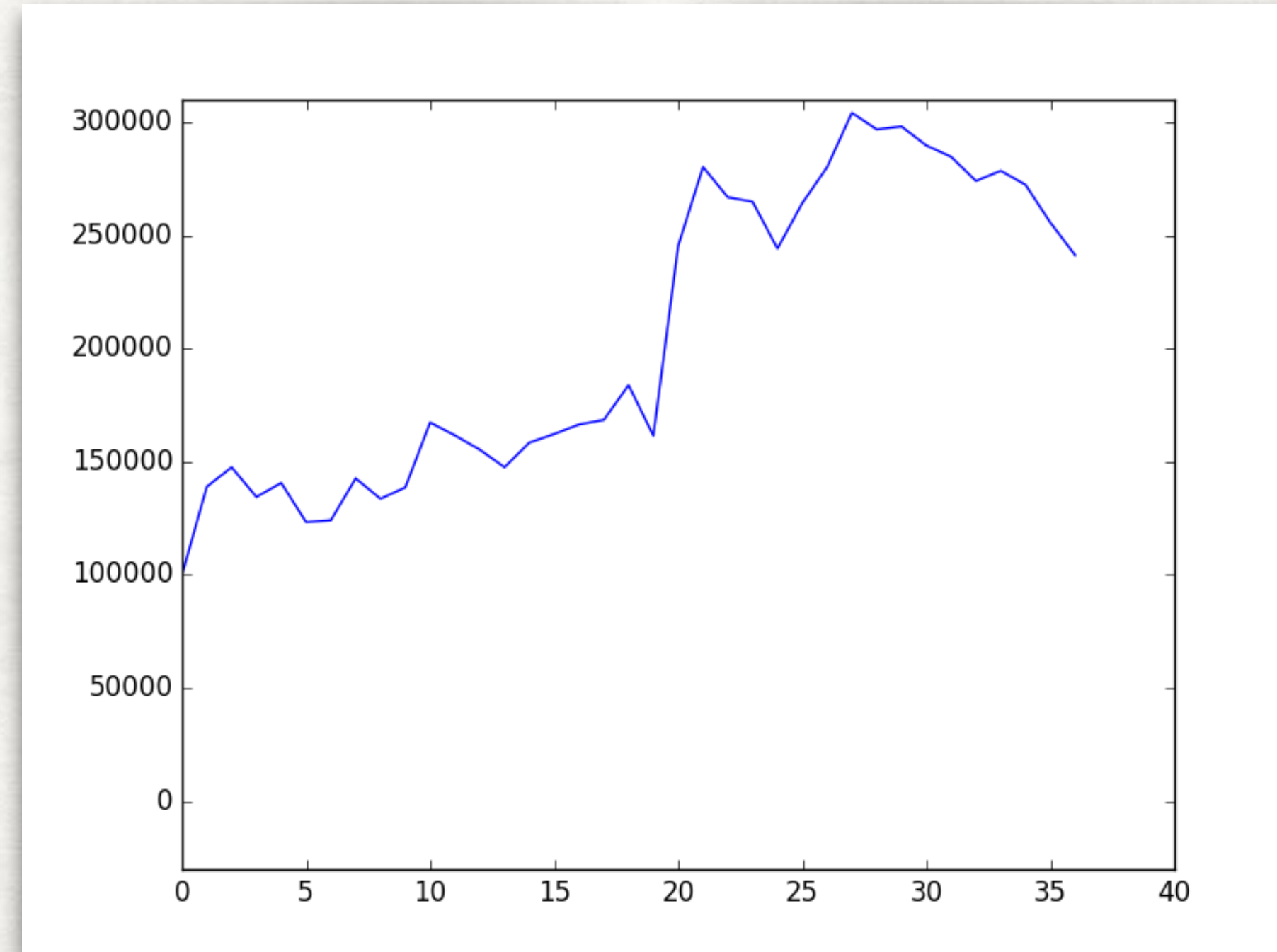


回収率 85%

的中率 49%

的中率は高いが回収率が低い

配当期待値予測モデル



回収率116%

的中率 26%

的中率は低いが回収率が高い