令和３年度　修士論文

包除積分を利用したディープラーニングネットワークの構築

令和３年２月

九州工業大学大学院情報工学府　学際情報工学専攻

システム創成情報工学分野

19677501

板橋　将之

指導教員：本田　あおい

目次

第1章 序論

1.1 研究背景と目的

近年、画像処理や自然言語処理などのデータ分析の分野において機械学習の手法の一つであるディープラーニングが使用され高い成果を上げている。ディープラーニングが用いられる理由としては複雑な特徴量を自動検出し高い精度を得ることがあげられる。しかし、このディープラーニングは複雑であるために解釈性が損なわれるという問題点があり、一般的に解釈性と精度はトレードオフの関係性にあると考えられている。解釈性が必要な例で言うと、例えば、ディープラーニングで学習した医療診断ソフトにより病気の診断をしたとしても、その原因となるものが何かわからないのでは対処のしようがないのである。そこで私の研究では説明可能でありながら高い精度も保持できるようなネットワークを考えた時に、1+1 が 2 にならない非加法的測度であるファジィ測度と多項演算を用いて定義される包除積分という積分式を用いてネットワークを構築することを提案し、実際に高い精度が得られるのか、どのような解釈が得られるのか実験を行った。

1.2 アプローチ

包除積分はファジィ測度からなる積分式であるがメビウス逆変換を行うことによって重回帰のような式を得ることができる。包除積分を利用したディープラーニングの構築を行うために、このメビウス逆変換の式を採用し、ディープラーニングネットワークの出力で得られる式が最終的に得られる包除積分式となるようにネットワークを構築した。今回のネットワーク構築では、以前まで対応できていなかったデータに対しても正しく学習できるようにネットワークの層やユニット数を増やし、データへの汎用性を向上させた。解釈性についてもゲーム理論でよく使われるシャープレイ値をさらに拡張した 2 変数以上の相互作用関係まで見ることができる手法を取り入れさらなる解釈性の向上を目指した。また、提案手法の相対的な精度や解釈を見るためにデータセットの提供やコンペティションが行われている kaggle でよく使われている手法で、解釈もできるような手法である XGBOOST と比較する。実験に使用するデータは Python の機械学習ライブラリである scikit-learn 内に含まれている「diabetes」（日本語訳：糖尿病）に関するデータを用いることで糖尿病について予測しその原因について解釈していく。

第2章　数学的定義

本論文を通して Ω＝｛1,2,…,n｝は有限集合とする。
|A|は有限集合 A の要素の個数を、P(Ω)は Ω の部分集合からなる集合、つまり Ω のべき集合を表す。

2.1　包除積分の定義

2.1.1 包除積分

定義１（ファジィ測度[1,2]）Ω 上の集合関数 μ：P(Ω)→[0,∞]は
(i)μ(Ø)=0,
(ii)A,B∈P(Ω)で A⊂B ならば μ(A)≦μ(B)
を満たすとき、Ω 上のファジィ測度という。

定義２(t-ノルム)[0,K]上の二項演算⊗: $[0,K]^2$→　[0,K]は x,y,z∈[0,K]に対して、
(i)0⊗0=0, x⊗K=x,
(ii)x≦y implies x⊗z≦y⊗z,
(iii)x⊗y=y⊗x
(iv)(x⊗y)⊗z=x⊗(y⊗z)
を満たすとき t-ノルムという。

通常、t-ノルムは[0,1]上に定義されるのが一般的であるが本質的な違いはない、実際、[0,1]上の t-ノルム⊗は

$$x\otimes^{G}y:=\left(\frac{x}{K}\otimes\frac{y}{K}\right)K$$

とすることで簡単に[0,K]上の t-ノルムになり、このとき K が単位元となる。次に示す論理積、代数積、Dubois-Prade の t-ノルム、Dombi の t-ノルムの他にも多くの t-ノルムが提案されている。

論理積：　$x\wedge y:=\min(x,y)$,

代数積：$x\otimes^{P}y:=\frac{xy}{K}$,

Dubois-Prade：$x\otimes_{\lambda}^{DP}y:=\frac{xy}{\max(x,y,\lambda)},0\leqq\lambda\leqq K$,

Dombi: $x\otimes_{\lambda}^{Db}y:=\frac{1}{1+\left(\left(\frac{1}{x}-1\right)^{\lambda}+\left(\frac{1}{y}-1\right)^{\lambda}\right)^{\left(\frac{1}{\lambda}\right)}},\lambda\in(0,\infty)$,

性質(iv)により t-ノルムは自然に多項演算に拡張できる。例えば Dubois-Prade の t-ノルムは

$$\otimes^{DP}_{i\in A} x_i=\begin{cases} K, & A=\emptyset, \\ \bigwedge_{i\in A} x_i & x_i>\lambda\ ,i\in A, \\ \prod_{i:x_i<\lambda} \frac{x_i}{\lambda^{|\{i|x_i<\lambda\}|-1}}, & \text{otherwise.} \end{cases},$$

この掛け算型の演算、t-ノルムを用いて包除積分を定義する。被積分関数 f は Ω 上に定義された[0,K]に値をとる非負有界値関数である。今、Ω は n 個の要素から成る有限集合なので、Ω 上の関数は n 次元の数ベクトルで表せる。これを$f=(x_1, x_2,.. x_n)\in[0,K]^n$と書くことにする。

定義３（包除積分[4]）μ をファジィ測度、⊗を[0,K]上の t-ノルムとする。Ω 上の非負有界関数$f=(x_1, x_2,.. x_n) \in[0,K]^n$の μ と⊗による包除積分は次で定義される。

$$\otimes\int f\,d\mu := \sum_{A\in P(\Omega)\setminus\{\emptyset\}} M^{\otimes}(f|A)\,\mu(A),$$

ただし、

$$M^{\otimes}(f|A) := \sum_{B\in P(\Omega),B\supset A} (-1)^{|B\setminus A|} \otimes_{i\in B} x_i,$$

定義式に現れる足したり引いたりの項がたくさん出てくる$M^{\otimes}(f|A)$の計算は包除原理に基づくもので、これが包除積分の名前の由来である。Ω={1,2,3}のときの包除積分を書き下すと

$$\begin{aligned}\otimes\int f\,d\mu :=\ & (x_1 - x_1\otimes x_2 - x_1\otimes x_3 + x_1\otimes x_2\otimes x_3)\,\mu(\{1\}) \\ & +(x_2 - x_1\otimes x_2 - x_2\otimes x_3 + x_1\otimes x_2\otimes x_3)\,\mu(\{2\}) \\ & +(x_3 - x_1\otimes x_3 - x_2\otimes x_3 + x_1\otimes x_2\otimes x_3)\,\mu(\{3\}) \\ & +(x_1\otimes x_2 - x_1\otimes x_2\otimes x_3)\,\mu(\{1,2\}) \\ & +(x_1\otimes x_3 - x_1\otimes x_2\otimes x_3)\,\mu(\{1,3\}) \\ & +(x_2\otimes x_3 - x_1\otimes x_2\otimes x_3)\,\mu(\{2,3\}) \\ & +(x_1\otimes x_2\otimes x_3)\mu(\{1,2,3\})\end{aligned}$$

メビウスの反転公式を用いると包除積分は次のように別表現できる。

$$\otimes\int f\,d\mu = \sum_{A\in P(\Omega)\setminus\{\emptyset\}} \left(\otimes_{i\in A} x_i\right) m^{\mu}(A),$$

ただしm^{μ}は μ のメビウス変換：

$$m^{\mu}(A) := \sum_{B\subset A} (-1)^{|B-A|}\,\mu(B).$$

この別表現を３点集合 Ω={1,2,3}の場合で書き下すと

$$\begin{aligned}\otimes\int f\,d\mu :=\ & x_1 m^{\mu}(\{1\}) + x_2 m^{\mu}(\{2\}) + x_3 m^{\mu}(\{3\}) \\ & +(x_1\otimes x_2)\ m^{\mu}(\{1,2\}) + (x_1\otimes x_3)\ m^{\mu}(\{1,3\})\end{aligned}$$

$+(x_2 \otimes x_3)\ m^{\mu}(\{2,3\})$

$+(x_1 \otimes x_2 \otimes x_3)\ m^{\mu}(\{1,2,3\})$.

定義式に比べてすっきりとして見えるのは足したり引いたりの部分がメビウス変換に含まれているためであり、メビウス変換に含まれているためであり、メビウス変換も書き下してしまうと定義式と同程度に煩雑な式となる。こちらを定義として採用しなかったのは非加法的測度 μ がそのままの形で表れているほうが、より自然であると考えたためである。

2.1.2 単調性判別

集合関数 μ のメビウス変換m^{μ}が次の条件を満たすとき、μ は単調性を満たす。

任意の $A \in 2^{\Omega}$ に対して

$m^{\mu}(A) \geqq -\sum_{B \subset \neq A, B \ni i} m^{\mu}(B)$

が任意の$i \in A$に対して成り立つ。

2.1.3 シャープレイ値

包除積分で求められるファジィ測度は$2^n - 1$となり、Ωの要素数が少ない場合はファジィ測度からの解釈は比較的容易だが、Ωの要素数が多い場合に簡単にファジィ測度から各要因の貢献度などの比較を行うことは難しい。シャープレイ値はゲーム理論で使われている手法で、協力ゲームにおいてプレイヤーに貢献度に応じた報酬を分配する手法の一つである。このシャープレイ値をファジィ測度で計算することができ、各要因ごとの比較も可能となる。各要素のシャープレイ値は次式で計算できる。

$$\phi_i(\mu) = \sum_{A \in \Omega \setminus \{i\}} \frac{|A|!\,(n - |A| - 1)!}{n!} (\mu(A \cup \{i\}) - \mu(A))$$

$\Phi(\mu) = (\phi_1(\mu), \phi_2(\mu), \ldots, \phi_n(\mu))$の$\phi_i(\mu)$が要素$i$ の貢献度である。また、シャープレイ値は次のような性質がある。

1．個人合理性：$\phi_i(\mu) \geq \mu(\{i\})$

2．全体合理性：$\sum_{i \in \Omega} \phi_i(\mu) = \mu(\Omega)$

3．対称性：要素i, jが$\mu(S \cup \{i\}) \quad = \mu(S \cup \{j\}) \quad S: \Omega$ のi, jを含まないすべての部分集合の場合、$\phi_i(\mu) = \phi_j(\mu)$

4．加法性：2 つの特性関数vとwによって作られた協力ゲームの和$v + w$において、

$\phi_i(v + w) = \phi_i(v) + \phi_i(w)$

がΩのすべての要素に対してそれぞれ成り立つ。

5．ナルプレイヤーに関する性質：ナルプレイヤーとは、iがΩのiを含まないすべての部分集合Sについて、$\mu(S \cup \{i\}) = \mu(S)$

を満たすことを言い、このナルプレイヤーに対して報酬を与えない。

実験ではこのシャープレイ値を用いて各要素の比較を行っていく。

2.1.4 拡張型シャープレイ値

シャープレイ値は各要素の貢献度を示す値だが要素間の関係性を表すことはできていない。そこで、相互作用指標を計算できるようにしたのが次式である。

$$I(T) = \sum_{S \supseteq T} \frac{1}{|S|-|T|+1} m^{\mu}(S), \quad T \subseteq \Omega$$

$|T|=1$ の時$I(T)$はシャープレイ値となり、$|T| > 1$ の時、$I(T)$は相互作用指標となる。

2.2 ディープラーニング

2.2.1 パーセプトロン（単層ニューラルネットワーク）

ディープラーニングとは機械学習の一種で、生物のニューロンの働きをモデルとしたニューラルネットワークをコンピュータで処理したものである。ディープラーニングではニューロンの一つをユニット（もしくはノード）といい、ニューロンの部分をユニット、軸索の部分をエッジで表す。このユニットをいくつも繋げ入力層、隠れ層、出力層と複数の層にしていったものをディープラーニングという。

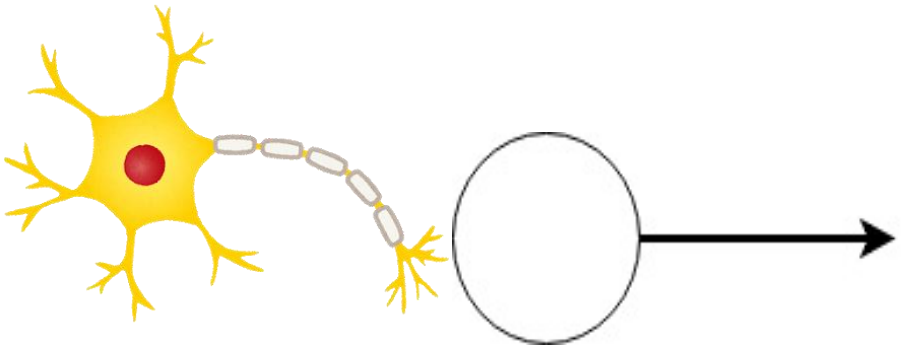


図 2.2.1　ニューロンとユニット

データを受け取る入力層と予測結果を出力する出力層の 2 層で構成されているものをパーセプトロンという。図 2.21 はパーセプトロンを示したものである。

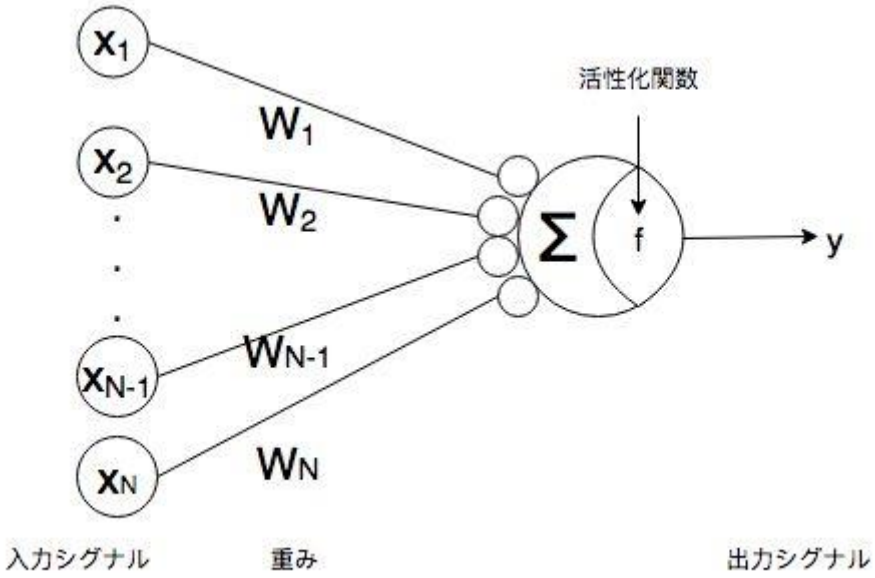


図 2.2.2　パーセプトロン

入力層と出力層の間のエッジには重みが与えられ、この重みを変えていくことにより出力で予測行う。図 2 のパーセプトロンの出力は $y=f(\sum_{i=1}^{N} x_i W_i)$となる。

このパーセプトロンを何層にもいくつも繋げたものを多層パーセプトロンという。ディープラーニングは多層パーセプトロンの重みをある手法で更新していくことで学習を行うものである。

2.2.2 活性化関数

活性化関数はニューロンの動きを模倣したもので、ある閾値を超えると発火して次のニューロンに電気信号を渡す働きを再現している。シグモイド関数、ReLU 関数、ステップ関数の他にも多数の活性化関数がある。

x は変数とする。

シグモイド関数：$\frac{1}{1+e^{-x}}$

ReLU 関数：$\max(0,x)$

ステップ関数：$\begin{cases} 1 & x>0 \\ 0 & x\leq 0 \end{cases}$

2.2.3 損失関数

損失関数は教師データと予測データの誤差を最小にするため重みやバイアスの値を最適化する仕組みを担うものである。この損失関数はディープラーニングで出力した予測値と実際の値との誤差全体を表す。今回実験に使用した損失関数は平均二乗誤差である。

平均二乗誤差：$E=\frac{1}{N}\sum_{i=1}^{N}(t_i-y_i)^2$

E：誤差関数、N：データ数、t：教師データ、y：予測データ

2.2.4 勾配降下法

勾配降下法は誤差関数を最小にするような重み、バイアスを求める手法である。
勾配降下法はいくつも提案されており、最も一般的な確率的勾配降下法の形を示すと、

確率的勾配降下法：$w^{t+1}=w^{t}-\mu\frac{\partial E(w^{t})}{\partial w^{t}}$

w：重み、t：更新タイム、E：誤差関数、μ：学習係数
の形で表す。今回の実験では Adam と呼ばれる手法を用いている。
Adam

$$m_{t+1}=\beta_1 m_t+(1-\beta_1)\nabla E(w^{t})$$
$$v_{t+1}=\beta_2 v_t+(1-\beta_2)\nabla E(w^{t})^2$$
$$\hat{m}=\frac{m_{t+1}}{1-\beta_1^{t}}$$
$$\hat{v}=\frac{v_{t+1}}{1-\beta_2^{t}}$$
$$w^{t+1}=w^{t}-\alpha\frac{\hat{m}}{\sqrt{\hat{v}}+\varepsilon}$$

ただしm_0=0、v_0=0、α=0.001、 β_1=0.9、β_2=0.999、 $\varepsilon=10^{-8}$とする。
w：重み、t：更新タイム、α：学習率、m と vは初期値 0 の変数である。

2.2.5 バッチ処理

バッチ処理は更新に使用するデータ数を変えていく処理である。データ全体を使ってパラメータを更新することをバッチ学習という。データ一つを使ってパラメータを更新することをオンライン学習という。今回使用するのはこのバッチ学習とオンライン学習の中間であるミニバッチという手法である。ミニバッチはデータ全体をいくつかに等分したものでそれぞれ更新する手法である。Dをランダムに M 等分して、各D_iで学習する場合、

ミニバッチ：$D=\bigcup_{i=1}^{M} D_i$

となる。

2.2.6 スパースモデリング

スパースモデリングはデータが不足している場合でも、少量のデータから解析を可能とする手法である。大量のデータを使わないため解析時間を短縮でき、データの構造をわかりやすく表現することができる。

2.3 重回帰分析

重回帰分析とは一つの目的変数と多数の説明変数の間の関係を予測する分析である。各説明変数の偏回帰係数a_iと切片a_0を求めることによって目的変数 y を求める。下に重回帰式を示す。

重回帰式：$y=a_1x_1+a_2x_2+\ldots+a_nx_n+a_0$

2.4XGBOOST

2.5 その他機械学習

2.6 評価方法

回帰問題では損失関数や決定係数、赤池情報量などの評価方法があるがこの論文では評価方法として以下の 3 つを採用し、実験ではこの値を見ることによってモデルの性能を評価していく。

1：平均絶対誤差（Mean Absolute Error　:MAE）

MAE は教師データと予測データの絶対値を平均したもので、小さいほど誤差が少なく良い予測モデルが得られていることがわかる。計算式は以下のようになる。

平均絶対誤差：$E=\frac{1}{N}\sum_{i=1}^{N}|t_i-y_i|$

E：誤差関数、N：データ数、t：教師データ、y：予測データ

2：平均 2 乗誤差（Mean Squared Error　:MSE）

MSE は 2.2.3 で示したような式で、MAE に比べ大きな誤差がある場合に大きな値になり易い。実験の損失関数ではこれを用いて学習を行うが MAE と同じく誤差が小さいほど良い予測モデルが得られていることがわかる。

３：決定係数（R^2）

回帰モデルの評価に良く用いられる指標で、良い予測モデルの場合 1 に近い値をとる。計算式はいくつか存在するが、今回の実験では scikit-learn の sklearn.metrics.r2_score 関数で使われている計算式を用いる。計算式は以下のようになる。

決定係数：$R^2(t,y)=1-\frac{\sum_{i=1}^{N}(t_i-y_i)^2}{\sum_{i=1}^{N}(t_i-\bar{t})^2}$

E：誤差関数、N：データ数、t：教師データ、y：予測データ、$\bar{t}$：教師データの平均値

第3章　提案手法

3.1 メビウス型包除積分モデル１

包除積分はメビウス変換で重回帰形に式変形をすることができるが、そのメビウス変換の値を求めることが課題となっている。この研究ではニューラルネットワークによる重みの計算でそのメビウス変換の値を求め、包除積分を得ることでモデルの解釈を行っていくことを目的としている。実際に作成するモデルはデータの入力層、データの前処理を行う層、包除積分の式を表す層、そして出力層の大きく分けて 4 層からなるネットワークモデルとなる。図６？は入力を 3 変数としたときのネットワークモデルである。図６？では入力層の各ユニットの説明変数x_1～x_3それぞれが前処理層のそれぞれのユニットにエッジを持ち、前処理層では活性化関数として sigmoid 関数を用いることで、各説明変数を 0～1 の間の値にすることができる。この前処理によって各説明変数を 0～1 の間にすることで t-ノルムの定義である x,y,z∈[0,K]を x,y,z∈[0,1]にし、代数積はただの積の多項演算とすることができる。包除積分を表す層では、前処理層によって 0～1 になった値を受け取り、t-ノルムの演算を行った後、活性化関数を恒等関数として出力し、その出力と重みの積を出力層に渡す。この時の重み$W_{\{1\}}$~$W_{\{1,2,3\}}$は多項演算との積、つまりメビウス変換である$\mathrm{m}^{\mu}(\{1\})$～$\mathrm{m}^{\mu}(\{1,2,3\})$に相当するため、この重みを求めることで包除積分を得ることができる。出力層では回帰問題や分類問題によって活性化関数を変えることができ、この y の値でデータに対する予測を行う。このモデルの特徴としては、説明変数を前処理によって 0~1 の sigmoid 関数の波形として得るため、目的変数と説明変数の相関関係からある程度初期値を予測することが可能であり、適切な初期値を与えることで学習の収束にかかる時間を早めることができる。実験ではそれぞれの各変数のユニットから前処理層につながっているエッジの重みをa_m(m=1,2,…説明変数の数),バイアスをb_mとして、各説明変数データの最大値、最小値を $\max(\boldsymbol{x}_m)$、$\min(\boldsymbol{x}_m)$として、目的変数に対する説明変数$\boldsymbol{x}_m$の相関が正ならば次の式で初期値を計算する。

$$a_m = \frac{6}{\max(\boldsymbol{x}_m) - \min(\boldsymbol{x}_m)}$$

$$b_m = \frac{-6(\max(\boldsymbol{x}_m) + \min(\boldsymbol{x}_m))}{2(\max(\boldsymbol{x}_m) - \min(\boldsymbol{x}_m))}$$

目的変数に対する説明変数$\boldsymbol{x}_m$の相関が負ならば、

$$a_m = \frac{-6}{\max(\boldsymbol{x}_m) - \min(\boldsymbol{x}_m)}$$

$$b_m = \frac{6(\max(\boldsymbol{x}_m) + \min(\boldsymbol{x}_m))}{2(\max(\boldsymbol{x}_m) - \min(\boldsymbol{x}_m))}$$

とただ符号を逆転した初期値を与える。このモデルの欠点としては、説明変数が出力に対して単調でない場合、例えば説明変数に対して 2 次関数の凸となるような sigmoid 関数では

表現できない波形が最適解だった場合に高い精度を出すことができないというような点がある。そのため、データとしては目的変数に対して相関の強いものを説明変数とすることが好まれる。

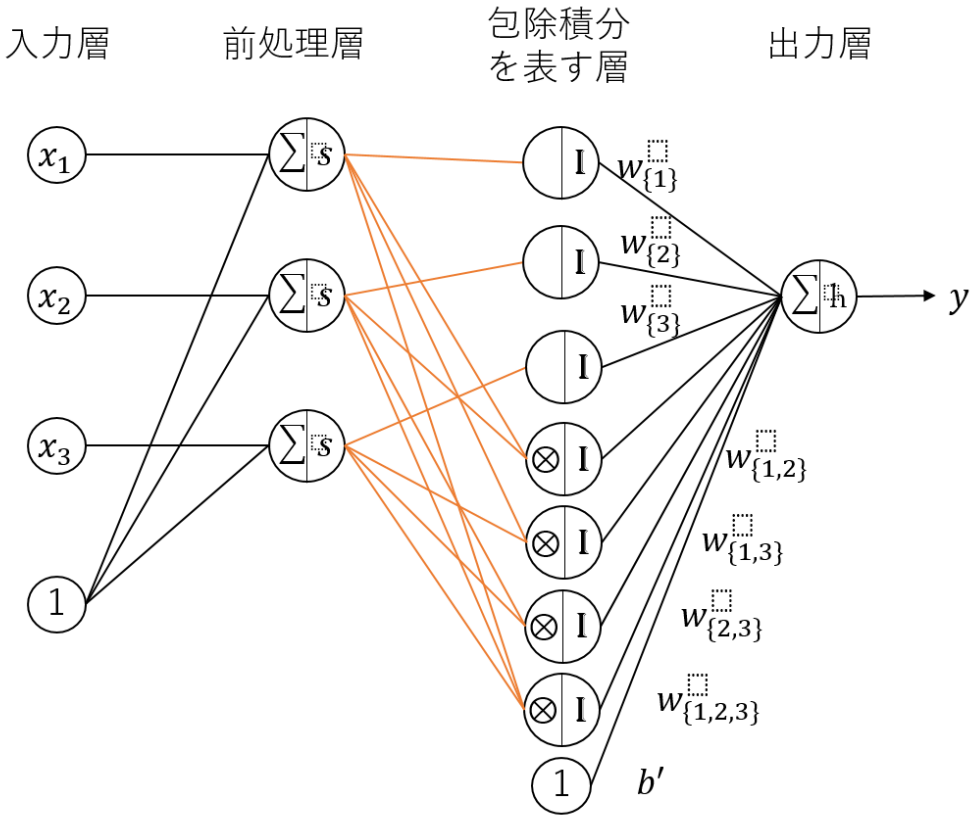


図６？

3.2 メビウス型包除積分モデル２

3.1 で述べた欠点を補うため入力層から前処理層の間に複数のユニットを用意し、各説明変数の特徴を学習するようにした。図７？は 3 入力としたときのメビウス型包除積分モデル２である。適当なユニット数とは図８？のようなネットワークを略したもので、各説明変数のデータによってどのような前処理を行えばよいかは異なるため初期値は予測が難しい。ユニット数は少なすぎると最適解を得られず、多すぎると学習に時間がかかるため適当なユニットは手動で調整する必要がある。このモデルの特徴は用意したユニットに対し適切な初期値を与えることができないため、3.1 のモデルより学習に時間がかかる。ただし、2 次関数の凸となるような波形であっても表現することができる。

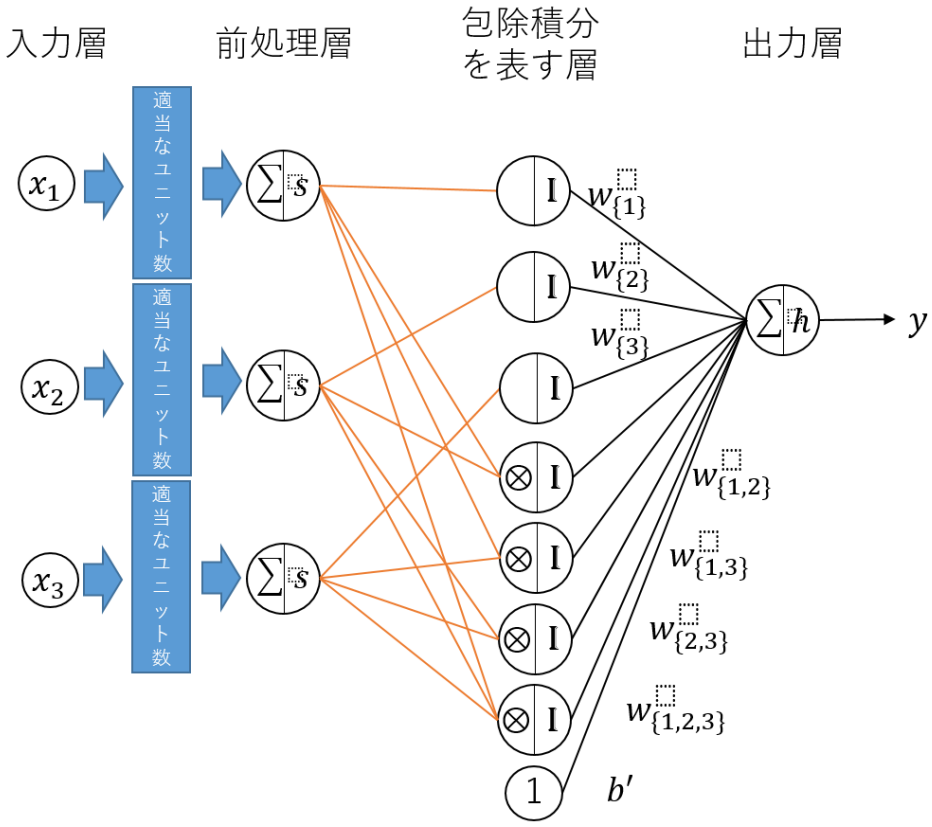

入力層
前処理層
包除積分
を表す層
出力層
適当なユニット数
適当なユニット数
適当なユニット数
x_1
x_2
x_3
s
s
s
I
I
I
I
I
I
I
$w_{\{1\}}$
$w_{\{2\}}$
$w_{\{3\}}$
$w_{\{1,2\}}$
$w_{\{1,3\}}$
$w_{\{2,3\}}$
$w_{\{1,2,3\}}$
1
b'
h
y

図７？

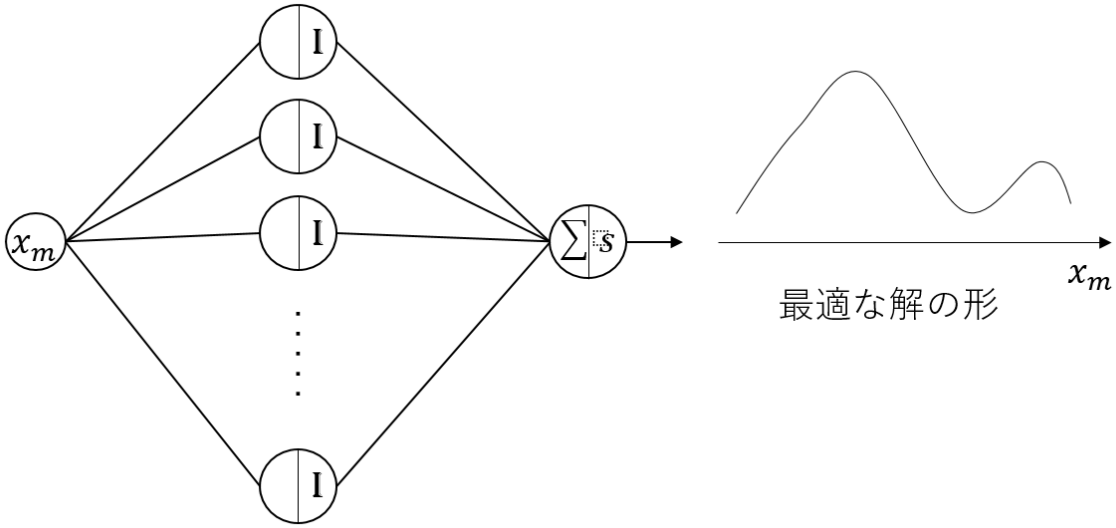

I
I
I
I
x_m
s
x_m
最適な解の形

図８？

第4章　実験

4.1 前準備

実験に使用するデータはPythonの機械学習ライブラリであるscikit-learnで提供されている「diabesets」（糖尿病）に関するデータを使用する。データラベルは表？？のようになっており、説明変数を年齢、性別、BMI値、平均血圧、総コレステロール、悪玉、善玉、血清に関する指標の計10として目的変数であるyの1年後の疾患進行度を予測するような回帰問題を解くデータとなっている。また、scikit-learnのライブラリから読み込む場合、すでに標準化されているため、
https://www4.stat.ncsu.edu/~boos/var.select/diabetes.tab.txt よりオリジナルデータを読み込み表？？のようにデータの詳細を見ていく。データ数は442で各変数の最大値、最小値が異なることがわかる。データの比較を行いやすくするためにデータの正規化を行い、表？？のようにした。正規化の式は以下のようにする。

$$x_{new} = \frac{x_{old} - \min(\boldsymbol{x})}{\max(\boldsymbol{x}) - \min(\boldsymbol{x})}$$

この時x_{new}が正規化後の説明変数の値で、x_{old}が正規化前の値、$\max(\boldsymbol{x})$、$\min(\boldsymbol{x})$はそれぞれ各説明変数データの最大値と最小値を示す。このようにすることですべての変数に対して0～1の値にすることができる。また、データの相関は図？？のようになった。図？？からldlとtc、tchとldlに高い正の相関があることが分かった。また、tchとhdlにも強い負の相関がみられた。これらのldl,tc,tch,hdlは多重共線性となる可能性があるが、実験1ではすべてのデータを用いて実験を行った。

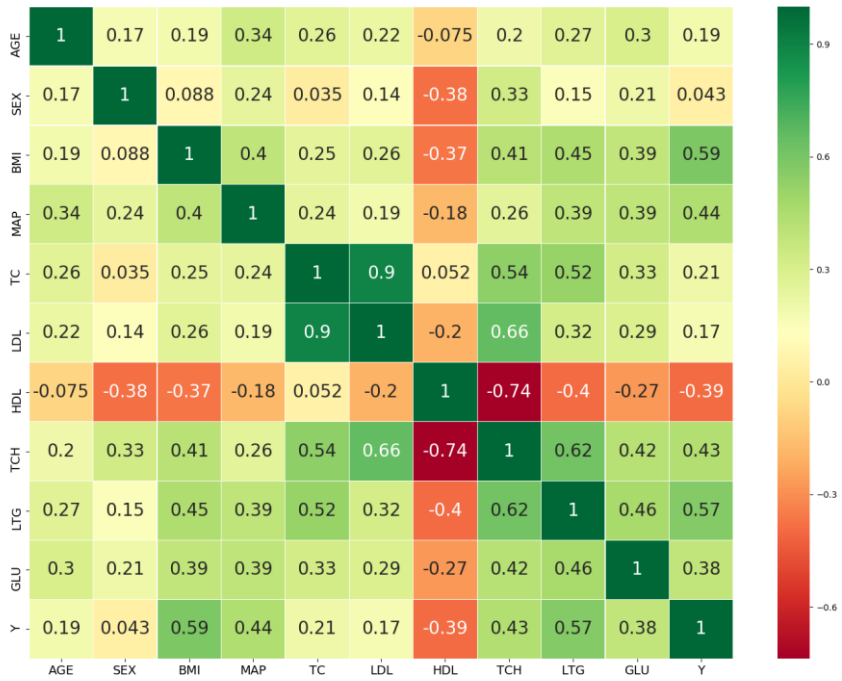
表

age	年齢
sex	性別
bmi	BMI値
map	平均血圧
tc	総コレステロール
ldl	悪玉
hdl	善玉
tch	血清に関する指標
ltg	
glu	
y	1年後の疾患進行度

表

	count	mean	std	min	25%	50%	75%	max
AGE	442.0	48.5	13.1	19.0	38.3	50.0	59.0	79.0

SEX	442.0	1.5	0.5	1.0	1.0	1.0	2.0	2.0
BMI	442.0	26.4	4.4	18.0	23.2	25.7	29.3	42.2
MAP	442.0	94.6	13.8	62.0	84.0	93.0	105.0	133.0
TC	442.0	189.1	34.6	97.0	164.3	186.0	209.8	301.0
LDL	442.0	115.4	30.4	41.6	96.1	113.0	134.5	242.4
HDL	442.0	49.8	12.9	22.0	40.3	48.0	57.8	99.0
TCH	442.0	4.1	1.3	2.0	3.0	4.0	5.0	9.1
LTG	442.0	4.6	0.5	3.3	4.3	4.6	5.0	6.1
GLU	442.0	91.3	11.5	58.0	83.3	91.0	98.0	124.0
Y	442.0	152.1	77.1	25.0	87.0	140.5	211.5	346.0



図

4.1.1　excel による単回帰分析

最初の回帰分析として excel による単回帰分析を行った。表？？はその分析結果である。表？？の回帰統計を見ると重決定　R2　が 0.5177、MAE が 0.1348、MSE が 0.0278 であることがわかる。重決定とは決定係数のことなので 0.5177 が決定係数となる。

表？？

回帰統計	
重相関 R	0.7195
重決定 R2	0.5177
補正 R2	0.5066
標準誤差	0.1687
観測数	442

	自由度	変動	分散	観測された分散比	有意 F
回帰	10	13.17	1.32	46.27	3.829E-62
残差	431	12.27	0.03		
合計	441	25.44			

	係数	標準誤差	t	P-値	下限 95%	上限 95%
切片	-0.084	0.097	-0.866	0.387	-0.276	0.107
AGE	-0.007	0.041	-0.168	0.867	-0.087	0.073
SEX	-0.071	0.018	-3.917	0.000	-0.107	-0.035
BMI	0.422	0.054	7.813	0.000	0.316	0.529
MAP	0.247	0.050	4.958	0.000	0.149	0.345
TC	-0.693	0.364	-1.901	0.058	-1.409	0.023
LDL	0.467	0.332	1.406	0.160	-0.186	1.120
HDL	0.089	0.188	0.475	0.635	-0.280	0.458
TCH	0.144	0.132	1.097	0.273	-0.114	0.403
LTG	0.608	0.139	4.370	0.000	0.334	0.881
GLU	0.058	0.056	1.025	0.306	-0.053	0.168

MAE	0.1348
MSE	0.0278

4.1.2 Support Vector Machine による分析

Support Vector Machine とは教師あり学習でよく用いられるパターン認識モデルの１つで回帰分析にも用いることができ、Support Vector Regression (SVR) として scikit-learn でもその学習用ライブラリが提供されている。単純な SVR モデル式は以下のようになる。

$$f(x_i) = w^T x_i + b$$

$f(x_i)$: SVR モデル　x_i:説明変数 w :重みベクトル b：バイアス

この重みとバイアスを求める式が以下の式である。

$$(w, b) = arg \min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_i max[|t_i - (w^T x_i + b)| - \varepsilon, 0]$$

x_i:説明変数 w :重みベクトル b：バイアス　t:教師データ

この式の C と ε が SVR のハイパーパラメータとなり、C は値が大きいほど過学習が起きやすく、 ε は教師データと予測データの誤差の絶対値が ε 以下であるデータを除くことで頑強なモデルにしようというパラメータである。3 つ目のハイパーパラメータはカーネル関数を選択するようなハイパーパラメータで、今回は線形カーネル、多項カーネル、RBF カーネルの 3 つで分析を行った。表？？はその結果である。表？？より SVM では MAE、MSE、R2 のどれでも RBF カーネルが良いという結果が得られた。

表？？

SVMによる分析結果（C=1, ε =0.1）			
カーネル関数 / 評価指標	線形カーネル	多項カーネル	RBFカーネル
MAE	0.1348	0.119	0.1113
MSE	0.0280	0.0227	0.0202
R2	0.5127	0.6054	0.6486

4.1.3 回帰木による分析

回帰木は決定木の一種で、木構造を用いることで回帰問題を解くことができる機械学習手法である。決定木では木構造が深くなればなるほど学習データに対して過学習を起こしてしまうため通常は木の深さを指定し学習を行う。表？？は木の最大の深さを変えた場合の評価指標を示している。表？？では木の深さを深くするにつれて精度が高くなっていることがわかり一見よさそうに見えるが、これは学習データに対する精度であって多くの場合テストデータを作ると過学習していることが多い。そのため決定木では次の分析で使うランダムフォレストなどで過学習を防いでいる。

表？？

回帰木							
木の深さ / 評価指標	3	4	5	7	10	15	20
MAE	0.1377	0.1251	0.1115	0.0698	0.0206	2.27E-05	0
MSE	0.0287	0.0244	0.0196	0.0104	0.002	0.0009	0
R2	0.5007	0.57561	0.6595	0.82	0.9645	0.9996	1

4.1.4 ランダムフォレストによる分析

ランダムフォレストとは学習データの一部を重複ありで取り出し、それぞれの異なるデータで学習した複数の異なる決定木から多数決を取るような機械学習手法である。この多数決を取ることで一つの決定木では陥りやすかった過学習を防ぐことができる。表？？は決定木 100 個用意し、木の最大の深さを指定した場合の評価指標の値である。表？？の結果から木の深さを深くしても R2＝１となることはなく決定木一つと比較し過学習は起きにくいことがわかる。

表？？

ランダムフォレスト(決定木の数：100)							
木の深さ / 評価指標	3	4	5	7	10	15	20
MAE	0.1298	0.1187	0.0167	0.0831	0.0607	0.0542	0.0553
MSE	0.0251	0.0208	0.1063	0.0101	0.0056	0.0045	0.0047
R2	0.5639	0.6387	0.7103	0.825	0.9034	0.9189	0.918

4.1.5　XGBOOST による分析

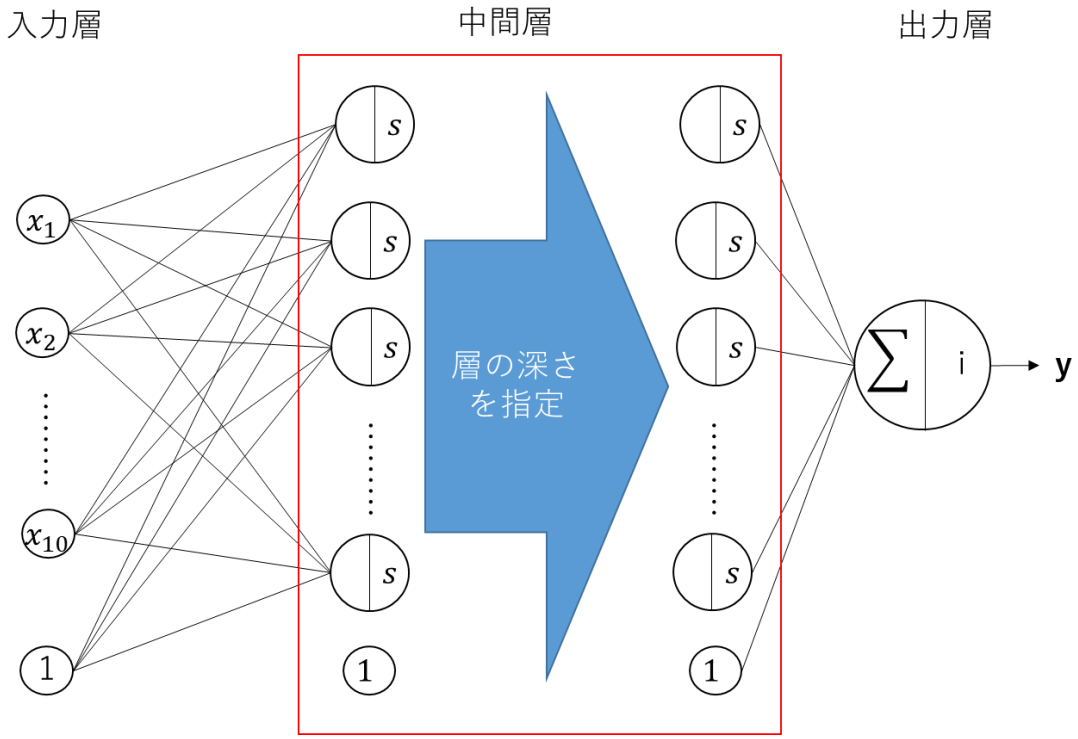
XGBOOST は多数の決定木を使うという点でランダムフォレストと似ているが、XGBOOST では一つの決定木で得られた誤差を小さくするような決定木をラウンド毎に追加していくという手法を取っている。ランダムフォレスト、XGBOOST はどちらもデータ分析を行う際によく使われる手法である。表？？は木の最大の深さを指定した場合の評価指標の値である。表？？の結果からランダムフォレストよりも学習データにフィットしやすいことがわかる。

表？？

XGBOOST(決定木の最大数：100)							
木の深さ / 評価指標	1	2	3	4	5	10	20
MAE	0.1183	0.0856	0.0461	0.0169	0.0039	0.0007	0.0004
MSE	0.0217	0.0119	0.0037	0.0005	3.16E-05	9.76E-07	3.27E-07
R2	0.623	0.7935	0.9351	0.99	0.9995	0.9999	1

4.1.6 NN による分析

提案手法の学習方法として使用している NN（ニューラルネットワーク）による分析も行っていく。NN モデルは層の深さやユニット数、活性化関数や誤差関数などを変えられるため、今回の実験では次の図のように指定した。



x :説明変数、s:シグモイド関数、i:恒等関数、y:予測値
中間層の層ごとのユニット数は 1000 とする

図？？

この NN モデルの中間層の層の数を変えながら実験を行った結果が表？？である。表？？の結果から～であることがわかる

表？？

4.2 実験 1

実験 1 では学習データに対して提案モデルがどの程度誤差を減少させることができるのかを調べるためテストデータを作らず全てのデータを学習データとし、提案手法以外にも機械学習手法で学習を行い決定係数、平均 2 乗誤差の二つの指標で比較を行っていく。使用するデータは 4.1 で前処理を行った diabesets データを使用する。表？？は 3.1 のメビウス型包除積分モデル 1 で一万回学習をさせた時の学習条件と決定係数、平均 2 乗誤差である。図？？は一万回の学習の平均 2 乗誤差を示している。

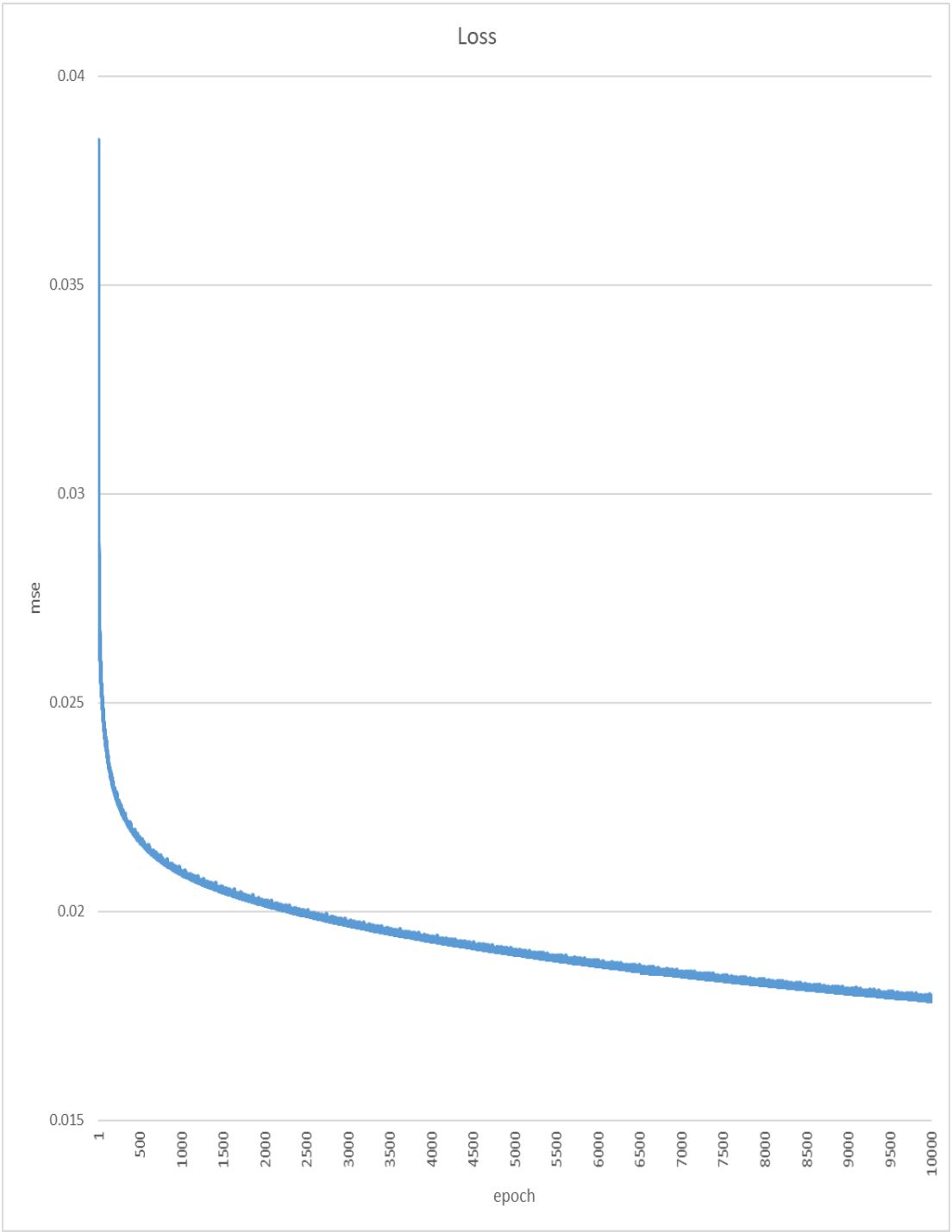
表？？

R2	MSE	MAE
0.6894	0.0179	

表

学習条件						
学習回数	バッチサイズ	更新式	t-norm	誤差関数	正則化項	サンプリン

						グ手法
1 万回	75	adam	代数積	平均 2 乗誤差	なし	なし



図？？：メビウス型包除積分モデル 1 の平均 2 乗誤差を示す学習曲線

また、提案手法では説明変数が 10 個の時は包除積分を表す層に存在するユニット数は2^{10}＝1024となる。これはすべての代数積を含むユニット数であるためこの代数積のユニットを減らすことによってよりスパースなモデルにすることができる。具体的には全 10 加法（$x_1 \otimes x_2 \sim x_1 \otimes x_2 \otimes x_3 \otimes x_4 \otimes x_5 \otimes x_6 \otimes x_7 \otimes x_8 \otimes x_9 \otimes x_{10}$まで）のところを 2 加法（$x_1 \otimes x_2 \sim x_5 \otimes x_6$まで）から 9 加法（$x_1 \otimes x_2 \sim x_2 \otimes x_3 \otimes x_4 \otimes x_5 \otimes x_6 \otimes x_7 \otimes x_8 \otimes x_9 \otimes x_{10}$まで）に制限を加えユニット数を減らす。図？？は 2 加法から全加法までの学習曲線である。

4.3 実験２

K 分割交差検証による汎化性能の比較

4.4 実験 3

シャープレイ値＋拡張したシャープレイ値の算出と考察 XGBOOST や単回帰分析との比較

4.5 実験 4

タイタニックデータによる 2 値分類で全データを使って実験

4.6 実験 5

タイタニックデータによる 2 値分類で 5 分割交差検証を使って実験

4.7 実験 6

実験 3 同様に重要度の比較を行っていく。

第5章　謝辞

本研究を行うにあたり、多くの貴重な御意見、御指導を賜りました本田　あおい准教授に心より感謝いたします。実験を行うにあたり、御協力を賜りました佐々木君、Alex 君をはじめとする本田研究室の皆様に心より感謝いたします。最後になりましたが、これまで私を見守り、支え続けてくれた家族に心より感謝します。

第6章　参考文献

第7章　データ