SUATU EVALUASI EKSPERIMENTAL TERHADAP *INCREMENTAL PROJECTION GENERALIZING NEURAL NETWORKS* PADA RUANG POLINOMIAL TRIGONOMETRI

Hendri Murfi* dan Benyamin Kusumoputro**
*Jurusan Matematika FMIPA - Universitas Indonesia, Depok 16424
e-mail: hendri@makara.cso.ui.ac.id
**Fakultas Ilmu Komputer - Universitas Indonesia, Depok 16424
e-mail: kusumo@cs.ui.ac.id

ABSTRAK

Salah satu hal penting dari suatu metode pembelajaran pada jaringan saraf tiruan adalah kapabilitas generalisasi. Yaitu kemampuan untuk memberikan hasil yang akurat terhadap data yang tidak diajarkan pada tahap pembelajaran. Salah satu metode pembelajaran incremental yang memberikan jaminan secara teori dalam menghasilkan kapabilitas generalisasi yang optimal adalah *incremental projection learning*. Pada tulisan ini akan dilakukan suatu evaluasi eksperimental terhadap jaringan saraf tiruan berbasis *incremental projection learning* ini yang dikembangkan pada ruang polinomial trigonometri dalam memecahkan masalah aproksimasi fungsi. Jaringan ini dikenal dengan nama *incremental projection generalizing neural networks*. Evaluasi dilakukan terutama pada masalah pemilihan model yang optimal dan jumlah neuron pada lapis tersembunyi ketika kapabilitas generalisasi optimal diperoleh. Selain itu, juga akan dilakukan suatu studi komparasi kapabilitas generalisasi yang dihasilkan oleh jaringan ini dengan kapabilitas generalisasi yang dihasilkan oleh jaringan lain yang sudah umum digunakan, yaitu *on-line backpropagation networks* dan *growing radial basis function networks*.

Kata kunci: *supervised learning*, *incremental projection learning*, kapabilitas generalisasi, jaringan saraf tiruan, masalah aproksimasi fungsi.

1. PENDAHULUAN

Salah satu kemampuan penting dari suatu metode pembelajaran pada jaringan saraf tiruan adalah kapabilitas generalisasi (*generalization capability*). Yaitu kemampuan untuk memberikan hasil yang akurat terhadap data yang tidak diajarkan selama proses pembelajaran. Kemampuan ini sangat penting terutama karena jumlah data pembelajaran yang menggambarkan fungsi target pembelajaran umumnya sedikit atau terbatas. Pada banyak persoalan dengan menggunakan jaringan saraf tiruan (JST), kita sering menemui kondisi dimana kapabilitas generalisasi ini diharapkan dapat terus ditingkatkan setelah proses pembelajaran selesai dilakukan. Misalnya ketika kita menemukan data dengan karakteristik baru yang berbeda dengan data yang pernah kita ajarkan ke jaringan. Salah satu pendekatan yang sering dilakukan untuk melakukan ini adalah dengan cara mengajarkan data baru tersebut ke jaringan dan jaringan akan menyimpan data tersebut sebagai informasi baru. Teknik pembelajaran seperti ini disebut juga dengan istilah *incremental learning* atau *sequential learning* atau *on-line learning*. Pendekatan seperti ini jika ditinjau dari sudut pandang pembelajaran pada diri manusia adalah sesuatu hal yang alami. Yaitu membangun pengetahuan berdasarkan pengetahuan sebelumnya. Disamping itu, metode pembelajaran yang bersifat *incremental* ini akan memainkan peranan yang sangat penting ketika kita bekerja pada pembelajaran yang bersifat aktif (*active learning*) [12], yaitu kita dapat memilih data pembelajaran selama proses pembelajaran. Sebaliknya, metode pembelajaran yang bersifat *batch* memerlukan semua data pembelajaran dalam proses pembelajarannya. Sehingga jika kita ingin meningkatkan kapabilitas generalisasi jaringan setelah proses pembelajaran selesai maka kita harus menyimpan semua data yang pernah diajarkan ke jaringan. Cara seperti ini jelas tidak efisien karena disamping harus menyimpan semua data pembelajaran, jaringan harus selalu direkon-struksi ulang setiap ada data pembelajaran yang baru. Akan tetapi kapabilitas generalisasi yang dihasilkan oleh metode pembelajaran yang bersifat batch ini umumnya lebih baik dari kapabilitas generalisasi yang dihasilkan oleh metode pembelajaran yang bersifat incremental

Dewasa ini telah banyak JST yang dikembangkan dengan menggunakan metode pembelajaran yang bersifat incremental dan umumnya menggunakan arsitektur jaringan multi lapis dengan satu lapis tersembunyi. Sebagian besar dilakukan dengan cara mengalokasikan neuron baru pada lapis tersembunyi ketika data pembelajaran baru diajarkan ke jaringan. Selanjutnya menentukan nilai bobot yang sesuai. Pada pendekatan seperti ini, jumlah neuron pada lapis tersembunyi tersebut akan terus meningkat seiring dengan pertambahan data pembelajaran. Untuk mencegah supaya neuron ini tidak terlalu banyak, Platt [5] mengenalkan suatu kriteria yang disebut *novelty criteria* dan jaringan yang dihasilkan disebut *resource allocating networks* (RAN). Kadirkamanathan

dan Nirajan [3] kemudian membuat suatu interpretasi RAN dari sudut pandang analisa fungsional dan menunjukkan bahwa akurasi yang diperoleh menjadi lebih baik. Selanjutnya, Yingwei et al [13] mengkombinasikan RAN versi Kadirkamanathan dan Nirajan diatas dengan suatu prosedur *pruning* yang akan membuang neuron-neuron yang memberikan kontribusi kecil pada jaringan dan jaringan yang dibentuk disebut *minimal resource allocating networks* (M-RAN). Walaupun RAN dan beberapa versi perbaikan seperti yang sudah dijelaskan diatas telah menunjukkan perbaikan dari sisi efisiensi komputasi, akan tetapi kapabilitas generalisasi yang optimal tidak dijamin secara teori. Hal ini disebabkan karena masing-masing jaringan tersebut hanya meminimalkan error pada data pembelajarannya saja. Sementara error pada data yang tidak diajarkan tidak menjadi perhatian metode ini.

Salah satu metode pembelajaran yang memberikan jaminan secara teori bahwa kapabilitas generalisasi yang diperoleh optimal adalah *incremental projection learning* (IPL) yang dikembangkan oleh Sugiyama dan Ogawa [8][9]. Sugiyama dan Ogawa juga menunjukkan bahwa kapabilitas generalisasi yang dihasilkan IPL ini sama dengan kapabilitas generalisasi yang dihasilkan oleh versi *batch*-nya, bahkan pada kasus non-asimtotik atau data yang kecil. IPL adalah versi incremental dari *projection learning* (PL) yang dikembangkan oleh Ogawa [4]. Ogawa merumuskan metode pembelajaran tersebut sebagai suatu masalah inverse (*inverse problem*) dengan pendekatan analisa fungsional. Pada metode ini, fungsi target pembelajaran dan fungsi hasil pembelajaran diasumsikan berada pada suatu ruang *reproducing kernel Hilbert*. Dalam mendapatkan fungsi hasil pembelajaran, metode ini akan mereduksi variansi dengan kondisi bias pada nilai minimal. Nilai minimal bias diperoleh dengan cara menempatkan fungsi hasil pembelajaran tersebut sebagai proyeksi orthogonal dari fungsi target. Dengan pendekatan seperti ini maka metode *projection learning* akan meminimumkan bias dan juga variansi dari fungsi hasil pembelajaran terhadap fungsi target pembelajaran. Kondisi inilah yang memberikan jaminan secara teori bahwa PL akan menghasilkan kapabilitas generalisasi yang optimal.

Pada tulisan ini, kami akan melakukan suatu evaluasi eksperimental terhadap JST berbasis IPL dalam memecahkan masalah aproksimasi fungsi pada ruang polynomial trigonometri sebagai ruang *reproducing kernel Hilbert*. JST ini dikenal dengan nama *incremental projection generalizing neural networks* (IPGNN). Evaluasi dilakukan terutama pada masalah pemilihan model yang optimal, yaitu pemilihan order basis dari ruang *reproducing kernel Hilbert* yang digunakan, dan jumlah neuron pada lapisan tersembunyi ketika kapabilitas generalisasi optimal diperoleh. Selain itu, kami juga akan melakukan studi komparasi kapabilitas generalisasi yang dihasilkan dengan kapabilitas generalisasi yang dihasikan oleh JST lain yang sudah umum digunakan, yaitu *on-line backpropagation networks* (BP) [6][2] dan *growing radial basis function networks* (RBF) [1][2]. Uji coba komputasi dilakukan dengan menggunakan perangkat lunak Matlab pada mesin Pentium II berbasis Windows.

Tulisan ini diorganisasikan sebagai berikut: Bagian 2 dan Bagian 3 akan menjelaskan tentang arsitektur dan algoritma IPGNN. Pada Bagian 4 akan diberikan hasil uji coba komputasi dan simulasi komputer yang diperoleh dalam memecahkan suatu masalah penguji buatan. Analisa terhadap hasil-hasil tersebut juga dibahas pada bagian ini. Tulisan ini akan ditutup dengan suatu kesimpulan pada Bagian 5.

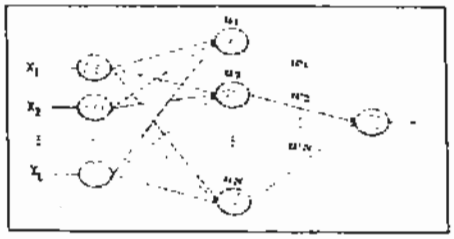
2. ARSITEKTUR IPGNN

IPGNN adalah suatu JST *feedforward* tiga lapis (*three-layer feedforward neural networks*) dengan arsitektur seperti pada Gambar 1. Arsitektur jaringan ini terdiri dari: lapis masukan, lapis tersembunyi dan lapis keluaran. Fungsi aktifasi hanya ada pada lapis tersembunyi dan menggunakan fungsi *reproducing kernel*. Pada arsitektur ini tidak ada bobot dari lapis masukan ke lapis tersembunyi. Hanya ada satu neuron pada lapis keluaran sementara pada lapis tersembunyi bersifat dinamis. Dimulai dengan tidak ada neuron pada awal pembelajaran, selanjutnya akan bertambah satu demi satu selama proses pembelajaran berdasarkan fungsi sampling dari data pembelajaran pada ruang aproksimasi yang direpresentasikan oleh lapis tersembunyi. Jika fungsi sampling dari data pembelajaran berada pada bidang aproksimasi atau fungsi reproducing kernel pada data pembelajaran tersebut bebas linear terhadap fungsi aktifasi yang ada pada jaringan maka satu neuron dengan fungsi reproducing kernel pada data pembelajaran tersebut sebagai fungsi aktifasi ditambahkan ke jaringan. Selanjutnya, bobot jaringan diperbarui. Sebaliknya, tidak ada penambahan neuron. Yang ada hanyalah memperbarui bobot yang ada. Bobot tersebut diperbarui dengan menggunakan kriteria IPL yang akan dibahas lebih rinci pada bagian selanjutnya.

Setelah tahap pembelajaran selesai, maka fungsi hasil pembelajarannya adalah kombinasi linear dari fungsi aktifasi yang ada dengan koefisiennya adalah nilai bobot yang bersesuai. Selanjutnya nilai aproksimasi dari suatu input (misal: x) yang dapat dirumuskan sebagai berikut:

$$f_m(x) = \sum_{i=1}^{N} w_i u_i(x) \qquad (1)$$

dimana x adalah vektor masukan, N adalah jumlah *neuron* pada lapis tersembunyi, w_i adalah bobot dari neuron ke-i, u_i adalah fungsi aktifasi pada *neuron* ke-i.



Gambar 1. Arsitektur Jaringan

3. ALGORITMA IPGNN

Dalam pengembangannya, masalah pembelajaran pada IPGNN dibagi dalam dua tahap. Tahap pertama adalah menentukan fungsi aproksimasi dari data pembelajaran dengan menggunakan algoritma IPL. Selanjutnya, JST yang merepresentasikan fungsi aproksimasi yang dihasilkan pada tahap pertama dibentuk pada tahap kedua dengan menggunakan algoritma *incremental projection learning in neural networks* (IPLNN) [7][11].

IPL adalah versi incremental dari metode *projection learning* yang dikembangkan oleh Ogawa. *Projection learning* dikembangkan dengan meminimumkan nilai generalization error yang dirumuskan sbb:

$$J_G = E_n\|f_m - f\|^2 \tag{2}$$

kemudian dikembangkan oleh Takemura menjadi

$$J_G = \|E_n f_m - f\|^2 + E_n\|f_m - E_n f_m\|^2 \tag{3}$$

dimana suku pertama disebut bias dan suku kedua disebut variansi dari f_m.

Projection learning dirumuskan oleh Ogawa sebagai suatu masalah inverse (*inverse problem*), dimana fungsi target dan fungsi hasil pembelajaran diasumsikan berada pada suatu ruang *reproducing kernel Hilbert* (H). Secara umum masalah *inverse* ini dapat dijelaskan sebagai berikut: misal $\{x_i, y_i\}_{i=1}^m$ adalah data pembelajaran, f adalah fungsi target, f_m adalah fungsi hasil pembelajaran dari m buah data pembelajaran, n_i adalah *noise* dari data pembelajaran ke-i, A_m adalah operator sampling (*sampling operator*) dari m buah data pembelajaran dan X_m adalah operator pembelajaran (*learning operator*) dari m buah data pembelajaran. Hubungan antara f dan $y^{(m)}$ ($=\{y_i\}_{i=1}^m$), $y^{(m)}$ dan f_m dapat diekspresikan sebagai

$$y^{(m)} = A_m f + n^{(m)} \tag{4}$$

$$f_m = X_m y^{(m)} \tag{5}$$

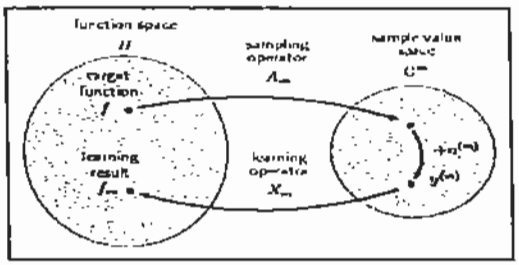
dimana A_m diekspresikan dengan menggunakan *Neuman-Schatten product* ($\otimes$)[1] sebagai

$$A_m = \sum (e_i^{(m)} \otimes \overline{\psi_i}) \tag{6}$$

dengan $e_i^{(m)}$ adalah vector berdimansi m dengan elemen nol kecuali elemen ke-i bernilai 1, ψ_i adalah fungsi sampling (*sampling function*) yang didefinisikan sebagai

$$\psi_i(x) = K(x, x_i) \tag{7}$$

dimana $K(x, x_i)$ adalah fungsi *reproducing kernel* pada x_i.



Gambar 2. Metode pembelajaran sebagai suatu masalah inverse (*inverse problem*)

Selanjutnya permasalahannya menjadi masalah inverse yaitu menentukan X_m yang akan memberikan aproksimasi terbaik f_m terhadap f berdasarkan suatu kriteria pembelajaran tertentu. Dalam menentukan X_m ini, Ogawa menggunakan kriteria pembelajaran proyeksi orthogonal. Dari Persamaan (4) dan (5), fungsi hasil pembelajaran f_m dapat didekomposisi sebagai

$$f_m = X_m A_m f + X_m n^{(m)} \tag{8}$$

Kemudian dari Persamaan (8) diatas, dapat dilihat bahwa

$$E_n f_m = X_m A_m f \tag{9}$$

dengan E_n adalah fungsi ekspektasi. Dari Persamaan (9) dapat dikatakan bahwa f_m merupakan anggota dari $R(X_mA_m)$, dimana $R(\cdot)$ menunjukkan *range* dari suatu operator. Misal P_S adalah operator proyeksi orthogonal pada suatu sub ruang S, maka untuk meminimumkan bias dari f_m, X_mA_mf haruslah merupakan proyeksi orthogonal dari f pada $R(X_mA_m)$:

$$X_m A_m f = P_{R(X_m A_m)} f \tag{10}$$

Karena semakin besar $R(X_mA_m)$ akan memberikan hasil aproksimasi yang lebih baik, Ogawa menggunakan yang terbesar yaitu:

$$R(X_m A_m) = R(A_m^*) \tag{11}$$

dengan A^* menunjukkan operator adjoint dari A. Dalam hal ini $R(A_m^*)$ disebut ruang aproksimasi (*approximation space*). Selanjutnya untuk lebih mereduksi *generalization error*, maka variansi dari f_m juga diminimalkan dengan tetap mempertahankan kondisi pada Persamaan (10) dan $R(X_mA_m) = R(A_m^*)$. Metode pembelajaran dengan pendekatan seperti ini selanjutnya disebut *projection learning*.

[1] Untuk sembarang g di ruang Hilbert H_1 dan sembarang f di ruang Hilbert H_2, *Neuman-Schatten product* (f⊗g) adalah operator dari H_1 ke H_2 yang didefinisikan dengan menggunakan sembarang h di H_1 sebagai $(f \otimes \overline{g})\, h = (h,g)\, f$.

Definisi 1 Suatu operator X_m disebut operator *projection learning* jika X_m meminimalkan secara fungsional

$$J_P[X_m] = E_n\left\|X_m n^{(m)}\right\|^2 \tag{12}$$

dengan kondisi

$$X_m A_m = P_{R(A_m^*)} \tag{13}$$

Jika Persamaan (8) dan (10) kita substitusikan ke Persamaan (3), maka akan didapat

$$J_G = \left\|P_{R(A_m^*)} f - f\right\|^2 + E_n\left\|X_m n^{(m)}\right\|^2 \tag{14}$$

Persamaan (14) menunjukkan bahwa *projection learning* akan meminimumkan variansi dan juga bias dari *generalization error*. Hal inilah yang menjamin secara teori bahwa *projection learning* akan memberikan *generalization error* atau kapabilitas generalisasi yang optimal.

Selanjutnya nilai X_m yang diperoleh, jika kita substitusikan ke Persamaan (5) maka kita akan mendapatkan fungsi hasil pembelajaran f_m. Dalam ruang Hilbert kita tidak dapat menentukan nilai fungsi pada suatu titik, akan tetapi jika ruang Hilbert tersebut memiliki suatu fungsi *reproducing kernel* maka nilai suatu fungsi pada suatu titik dapat ditentukan sebagai:

$$f_m(x) = \langle f_m, \psi(x)\rangle \tag{15}$$

dimana $\psi(x)$ adalah nilai fungsi sampling dari x pada ruang aproksimasi. Misal dimensi dari ruang aproksimasi adalah μ dan memiliki basis (fungsi *reproducing kernel*) $\{u_k\}_{k=1}^{\mu}$ maka

$$\psi(x) = (u_1(x), u_2(x), \ldots u_\mu(x)) \tag{16}$$

kemudian jika kita misalkan $f_m = (b_1, b_2, \ldots. b_\mu)$ dan *inner product* yang digunakan adalah *Euclidean inner product* (Ogawa telah menunjukkan bahwa pada *sample value space*, *Euclidean inner product* dapat digunakan tanpa kehilangan kapabilitas generalisasi) maka

$$f_m(x) = \sum_{k=1}^{\mu} b_k u_k(x) \tag{17}$$

Selanjutnya, Persamaan (17) diatas direpresentasikan dalam bentuk JST (Gambar 1) dimana fungsi aktifasinya adalah fungsi reproducing kernel dan bobotnya adalah b_k, yaitu fungsi hasil pembelajaran yang dihasilkan algoritma *projection learning*.

Algoritma *Projection learning*, seperti yang dijelaskan pada bagian sebelum, selanjutnya dikembangkan oleh Sugiyama dan Ogawa untuk pembelajaran yang bersifat *incremental* dan disebut IPL. Pada pembelajaran yang bersifat *incremental* ini, fungsi hasil pembelajaran (misal f_m) diperoleh dari fungsi hasil pembelajaran sebelumnya (misal f_{m-1}). Berikut ini adalah fungsi hasil pembelajaran dari salah satu algoritma IPL yang dikembangkan oleh Sugiyama dan Ogawa untuk kondisi dimana variansi *noise* positip.

Proposisi 1 Jika matrik korelasi *noise* adalah positip definit dan diagonal, fungsi hasil pembelajaran f_{m+1} diperoleh dari fungsi hasil pembelajaran sebelumnya f_m sebagai berikut

$$f_{m+1} = \begin{cases} f_m + \dfrac{\beta'_{m+1} V'_m \psi_{m+1}}{v_2^{(m+1)}} & jika\ \psi_{m+1} \in R(A_m^*) \\ f_m + \dfrac{\beta'_{m+1} \tilde{\psi}_{m+1}}{v_1^{(m+1)}} & jika\ \psi_{m+1} \notin R(A_m^*) \end{cases} \tag{18}$$

dimana

$$V'_m = A_m^* Q_m^{-1} A_m \tag{19}$$

$$\beta'_{m+1} = y_{m+1} - f_m(x_{m+1}) \tag{20}$$

$$\tilde{\psi}_{m+1} = P_{N(A_m)} \psi_{m+1} \tag{21}$$

$$v_1^{(m+1)} = \tilde{\psi}_{m+1}(x_{m+1}) \tag{22}$$

$$v_2^{(m+1)} = \sigma_{m+1} + \left\langle V'_m \psi_{m+1}, \psi_{m+1} \right\rangle \tag{23}$$

dengan Q_m adalah matrik korelasi *noise* dan σ adalah variansi *noise*.

Langkah berikutnya (tahap kedua) adalah membangun jaringan saraf tiruan yang merepresentasikan fungsi hasil pembelajaran yang dihasilkan oleh algoritma IPL. Algoritma yang dibuat untuk membangun jaringan saraf tiruan yang merepresentasikan fungsi hasil pembelajaran yang dihasilkan oleh algoritma IPL dikenal dengan nama *incremental projection learning in neural network* (IPLNN). Ada beberapa algoritma IPLNN yang sudah dikembangkan oleh Sugiyama dan Ogawa. Pada simulasi komputer yang akan kami lakukan, kami menggunakan algoritma IPLNN4 yang merepresentasikan fungsi hasil pembelajaran pada Persamaan (18) untuk kondisi dimana variansi noise data pembelajaran positip ($\sigma > 0$). (Gambar 3).

Kondisi $\psi_{m+1} \notin R(A_m^*)$ artinya bahwa ψ_{m+1} bebas linear terhadap $\{\psi_j\}_{j=1}^{m}$ dan ruang aproksimasi $R(A_{m+1}^*)$ menjadi lebih luas dari $R(A_m^*)$. Sebaliknya, $\psi_{m+1} \in R(A_m^*)$ artinya bahwa ψ_{m+1} tidak bebas linear $\{\psi_j\}_{j=1}^{m}$ dan ruang aproksimasi $R(A_{m+1}^*)$ sama dengan $R(A_m^*)$. Kondisi $\psi_{m+1} \notin R(A_m^*)$ dapat diperiksa karena $\psi_{m+1} \notin R(A_m^*)$ jika dan hanya jika

$$P_{N(A_m)} \psi_{m+1} = \tilde{\psi}_{m+1} \neq 0 \tag{24}$$

Selanjutnya, jika dimensi dari ruang *Reproducing Kernel Hilbert* yang digunakan adalah μ, maka kondisi $N(A_\mu) = \{0\}$ akan diperoleh setelah data pembelajaran yang memenuhi kondisi $\psi_{m+1} \notin R(A_m^*)$ ditambahakan sebanyak μ kali. Sehingga secara teori maka IPGNN akan selalu memiliki neuron pada lapisan tersembunyi lebih kecil atau sama dengan dimensi dari ruang *Reproducing Kernel Hilbert* yang digunakan. Kondisi ini sangat efisien karena pada banyak aplikasi, jumlah data pembelajaran

jauh lebih besar dari dimensi optimal ruang reproducing kernel Hilbert yang digunakan.

Dalam implementasinya, digunakan kriteria berikut:

Jika $\|\tilde{\psi}_{m+1}\|^2 = v_1 > \varepsilon$ maka $\psi_{m+1} \notin R(A_m^*)$ (25)

dimana ε adalah suatu konstanta yang kecil, misal $\varepsilon = 10^{-4}$

---

input $(x_{m+1}, y_{m+1}), \sigma_{m+1}$

if $m = 0$ {

generate first neuron in hidden layer

$u_1 \leftarrow K(x, x_1)$; $\quad w_1 \leftarrow \dfrac{y_1}{\|\psi_1\|^2}$;

$C_1^{(1)} \leftarrow \dfrac{\sigma_1}{\|\psi_1\|^4}$; $\quad D_1^{(1)} \leftarrow \dfrac{1}{\|\psi_1\|^2}$;

$N \leftarrow 1$;

} *else* {

$[b_N^{(m+1)}]_i \leftarrow \langle u_i, \psi_{m+1} \rangle$;

$[c_N^{(m+1)}]_i \leftarrow \sum_{j=1}^{N} [C_N^{(m)}]_{ij} \overline{[b_N^{(m+1)}]_j}$;

$[d_N^{(m+1)}]_i \leftarrow \sum_{j=1}^{N} [D_N^{(m)}]_{ij} \overline{[b_N^{(m+1)}]_j}$;

$\beta_{m+1} \leftarrow y_{m+1} - \sum_{i=1}^{N} w_i [b_N^{(m+1)}]_i$;

$v_1^{(m+1)} \leftarrow \psi_{m+1}(x_{m+1}) - \sum_{i=1}^{N} [c_N^{(m+1)}]_i [b_N^{(m+1)}]_i$;

$v_2^{(m+1)} \leftarrow \sigma_{m+1} + \sum_{i=1}^{N} [d_N^{(m+1)}]_i [b_N^{(m+1)}]_i$;

if $v_1^{(m+1)} \neq 0$ {

add the $(N+1)$*-st neuron in hidden layer*

$u_{N+1} \leftarrow K(x, x_{m+1})$, $w_{N+1} \leftarrow \dfrac{\beta_{m+1}}{v_1^{(m+1)}}$;

$w^{(N)} \leftarrow w^{(N)} - \dfrac{\beta_{m+1}}{v_1^{(m+1)}} c_N^{(m+1)}$;

$C_{N+1}^{(m+1)} \leftarrow \Gamma_{N+1} C_N^{(m)} \Gamma_{N+1}^* + \dfrac{a_{N+1}^{(m+1)} \otimes \overline{a_{N+1}^{(m+1)}}}{v_1^{(m+1)}}$

$D_{N+1}^{(m+1)} \leftarrow \Gamma_{N+1} C_N^{(m)} \Gamma_{N+1}^* + \dfrac{v_2^{(m+1)} (a_{N+1}^{(m+1)} \otimes \overline{a_{N+1}^{(m+1)}})}{v_1^{(m+1)^2}}$

$- \dfrac{\Gamma_{N+1} d_N^{(m+1)} \otimes \overline{a_{N+1}^{(m+1)}} + a_{N+1}^{(m+1)} \otimes \overline{\Gamma_{N+1} d_N^{(m+1)}}}{v_1^{(m+1)}}$;

$N \leftarrow N + 1$;

} *else* {

$w^{(N)} \leftarrow w^{(N)} + \dfrac{\beta_{m+1} d_N^{(m+1)}}{v_2^{(m+1)}}$;

$C_N^{(m+1)} \leftarrow C_N^{(m)}$;

$D_N^{(m+1)} \leftarrow D_N^{(m)} - \dfrac{d_N^{(m+1)} \otimes \overline{d_N^{(m+1)}}}{v_2^{(m+1)}}$;

}

}

Gambar 3. Algoritma IPLNN4

4. SIMULASI DAN ANALISA HASIL

Pada bagian ini akan diberikan hasil uji coba dan simulasi penggunaan IPGNN4 dalam memecahkan suatu fungsi matematika (Persamaan 26) dengan domain berada pada interval $[-\Pi, \Pi]$. Generalization error dihitung berdasarkan Persamaan (27).

$$f = 2x - 14e^{-3(x-2.5)^2} - 5e^{-6(x-0.5)^2} + 3e^{-2x^2} + 12e^{(x+2.5)^2} \quad (26)$$

$$Gen.err = \frac{1}{126} \sum_{n=0}^{125} [f(-\pi + 0.05n) - f_m(-\pi + 0.05n)]^2 \quad (27)$$

Ruang *reproducing kernel Hilbert* yang digunakan direntang oleh {1, sin kx, cos kx} dengan k adalah order basis. *Inner product* didefinisikan sebagai

$$\langle f, g \rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) \overline{g(x)} dx \quad (28)$$

Dan fungsi *reproducing kernel* didefinisikan sebagai

$$K(x, x') = \begin{cases} 2k+1 & jika\ x = x' \\ \dfrac{\sin \dfrac{(2k+1)(x-x')}{2}}{\sin \dfrac{x-x'}{2}} & jika\ x \neq x' \end{cases} \quad (29)$$

Pertama kami berikan simulasi pemilihan order basis untuk mendapatkan *generalization error* yang optimal (Tabel 1). Seperti karakteristik umum dari generalization error, pada simulasi ini juga terlihat bahwa generalization error akan menurun sampai suatu titik minimal kemudian akan berbalik naik. Pemilihan nilai order basis yang membuat nilai generalization error minimal inilah yang menjadi permasalah dalam pemilihan model (*model selection*) pada IPGNN. Akan tetapi, pada masalah riil generalization error ini tidak dapat dihitung secara langsung karena adanya fungsi target yang tidak diketahui. Pendekatan yang umum dilakukan adalah membuat suatu estimasi dari generalization error, kemudian memilih model yang meminimumkan nilai estimasi tersebut. Salah satu contoh estimasi tersebut adalah *subsapce information criteria* (SIC) [10]. Pada tulisan ini kami belum menggunakan suatu nilai estimasi dalam menentukan model yang optimal. Kami masih menggunakan fungsi generalization error seperti pada Persamaan (27).

Tabel 1. Simulasi pemilihan order basis yang optimal

Kasus	GE untuk Beberapa Nilai Order Basis						Order
	3	4	5	6	7	8	Optimal
I	2.0751	0.3584	0.5202	0.9794	1.5223	2.1645	4
II	2.9131	0.6271	0.3999	0.5978	1.1654	2.0772	5
III	4.2332	1.1236	0.9641	1.0927	1.8419	3.9111	5
IV	3.6541	1.2112	0.5292	0.4433	0.8453	2.9186	6
V	4.3223	1.1236	1.1037	1.4888	2.4902	3.9111	5

Kemudian akan disimulasikan pemilihan order basis yang optimal berdasarkan jumlah dan variansi *noise* dari data pembelajaran. Dari hasil simulasi yang diberikan

pada Tabel 2 terlihat bahwa order basis yang optimal sangat bergantung pada kedua hal tersebut. Dari hasil ini dapat dikatakan bahwa untuk mendukung supaya IPGNN dapat bekerja optimal maka dibutuhkan suatu metode pemilihan model yang incremental juga. Pemilihan model dengan menggunakan SIC yang dikembang oleh Sugiyama & Ogawa masih untuk skenario pembelajaran yang bersifat batch dan butuh pengembangan lanjutan untuk pembelajaran yang bersifat incremental.

Dari Tabel 2, kita juga melihat bahwa jumlah neuron pada lapisan tersembunyi cendrung bergantung pada jumlah data pembelajaran. Jumlah neuron tersebut tidak dibatasi oleh dimensi dimensi ruang *Reproducing Kernel Hilbert* yang digunakan. Hal ini tidak sesuai dengan teori yang mengatakan bahwa jumlah *neuron* tersebut akan lebih kecil atau sama dengan dimensi dari ruang *Reproducing Kernel Hilbert* yang digunakan.

Tabel 2. Simulasi jumlah neuron pada lapis tersembunyi pada saat kapabilitas generalisasi diperoleh

Kasus	Jumlah Data	Dist. *Noise*	Order Basis	Dim. Ruang	Jumlah Neuron
1	40	N(0,0.1)	6	13	40
2	40	N(0,0.2)	6	13	40
3	40	N(0,0.3)	6	13	40
4	40	N(0,1)	5	11	40
5	40	N(0,2)	5	11	40
6	40	N(0,3)	4	9	40
7	40	N(0,4)	4	9	40
8	30	N(0,0.1)	5	11	30
9	20	N(0,0.1)	4	9	20
10	10	N(0,0.1)	4	9	10
11	5	N(0,0.1)	4	9	5
12	30	N(0,1)	5	11	30
13	20	N(0,1)	4	9	20
14	10	N(0,1)	4	9	10
15	5	N(0,1)	4	9	5

Selanjutnya, kami akan melakukan studi komparasi terhadap kapabilitas generalisasi yang dihasilkan oleh IPGNN terhadap kapabilitas generalisasi yang dihasilkan oleh jaringan BP dan RBF. Karakteristik dari masing-masing JST tersebut adalah sebagai berikut:

- BP. Arsitektur jaringan terdiri dari tiga lapis. Jumlah neuron pada lapis tersembunyi merupakan model dari jaringan ini. Jumlah neuron yang dipilih pada simulasi adalah jumlah neuron yang membuat *generalization error* pada Persamaan (27) minimal. Disamping jumlah neuron ini, nilai inisialisasi juga mempengaruhi nilai generalization error. Sehingga nilai generalization error yang diberikan adalah nilai rata-rata untuk beberapa nilai inisialisasi. Fungsi aktifasi yang digunakan adalah fungsi *tag-sigmoid* pada lapisan tersembunyi dan fungsi linear pada lapisan output. *Epoch* yang diperlukan tidak lebih dari 3000 dan nilai *mean square error* tidak kurang dari 10^{-5}.
- RBF. Arsitektur jaringan terdiri dari tiga lapis. Jumlah neuron pada lapis tersembunyi bersifat dinamis. Mulai dari tidak ada pada awal pembelajaran selanjutnya berkembang selama proses pembelajaran. Model yang akan dioptimalisasi pada jaringan ini adalah nilai *spread factor* dari fungsi aktifasi. Fungsi aktifasi yang digunakan adalah fungsi *radial basis* pada lapisan tersembunyi dan fungsi linear pada lapisan output. Algoritma akan berhenti jika jumlah maksimal *neuron* pada lapis tersembunyi sudah tercapai atau nilai fungsi *sum square error* sudah lebih kecil dari 10^{-5}.

Tabel 3 adalah hasil simulasi terhadap 40 data pembelajaran dengan *noise* berdistribusi N(0,1). Dari Tabel 3 dapat dilihat bahwa IPGNN tidak selalu memberikan kapabilitas generalisasi yang terbaik. IPGNN dapat memiliki kapabilitas generalisasi terbaik (kasus I), pertengahan (kasus II) atau bahkan terburuk (kasus III).

Tabel 3. Tiga keadaan kapabilitas generalisasi dari IPGNN

Kasus	Jumlah Data	Dist. *Noise*	IPGNN	BP	RBF
1	40	N(0,1)	0.3999	0.8573	1.1943
2	40	N(0,1)	1.1037	0.8904	3.5178
3	40	N(0,1)	0.9641	0.6966	0.7221

Tabel 4. Perbandingan kapabilitas generalisasi berdasarkan jumlah data

Kasus	Jumlah Data	Dist. *Noise*	IPGNN	BP	RBF
1	40	N(0,1)	0.3999	0.7542	1.1943
2	30	N(0,1)	0.8083	0.7526	0.4486
3	20	N(0,1)	1.2895	0.7235	0.6550
4	10	N(0,1)	0.5823	4.3613	0.7283
5	5	N(0,1)	1.3314	6.4306	1.8085
6	40	N(0,0.1)	0.6017	0.2202	0.4931
7	30	N(0,0.1)	1.8677	0.3725	2.0004
8	20	N(0,0.1)	1.9294	0.3560	1.9839
9	10	N(0,0.1)	0.4918	3.9742	0.4646
10	5	N(0,0.1)	1.2998	8.9038	2.4679

Selanjutnya akan disimulasikan generalization error berdasarkan jumlah data pembelajaran. Tabel 4 merupakan nilai generalization error untuk beberapa jumlah data pembelajaran. Dari Tabel 4 terlihat bahwa kapabilitas generalisasi IPGNN cenderung lebih baik untuk data pembelajaran yang berukuran kecil, terutama jika

dibandingkan dengan BP. Simulasi grafis untuk Kasus 10 dengan lima data pembelajaran diberikan pada Lampiran pada Gambar L.1.

Pada Tabel 5 kita dapat melihat perbandingan kapabilitas generalisasi berdasarkan nilai variansi *noise* data pembelajaran. Nilai variansi yang diambil adalah nilai variansi yang cukup besar ($\sigma > 1$). Hasil yang diperoleh menunjukkan bahwa kapabilitas generalisasi dari IPGNN lebih baik dari BP maupun RBF. Simulasi grafis untuk Kasus 5 diberikan pada Lampiran pada Gambar L.2.

Tabel 5. Perbandingan
kapabilitas generalisasi untuk variansi *noise* cukup besar

Kasus	Jumlah Data	Dist. *Noise*	IPGNN	BP	RBF
1	40	N(0,2)	1.3102	3.2028	4.6839
2	40	N(0,2)	1.6034	2.5719	11.5226
3	40	N(0,2)	0.8587	2.5123	1.5664
4	40	N(0,3)	4.1020	7.1684	18.9112
5	40	N(0,3)	1.6403	4.5324	2.6182
6	40	N(0,3)	2.5363	4.7056	8.5409
7	40	N(0,4)	2.2749	8.1463	4.5663
8	40	N(0,4)	5.6786	7.9739	16.1306
9	40	N(0,4)	5.0104	8.6923	8.5244

Kasus terakhir yang akan diperbandingkan adalah kasus dimana data pembelajaran cukup besar dan nilai variansi *noise* cukup kecil. Tabel 6 menunjukkan nilai *generalization error* untuk data pembelajaran dengan kondisi tersebut. Dari Tabel 6 kita dapat melihat bahwa kapabilitas generalisasi dari IPGNN kurang baik jika dibandingkan dengan BP. Sementara terhadap RBF, IPGNN dapat lebih baik atau lebih jelek tergantung karakteristik data pembelajarannya. Simulasi grafis dari Kasus 4 diberikan pada Lampiran pada Gambar L.3.

Tabel 6. Perbandingan
kapabilitas generalisasi untuk variansi *noise* yang cukup kecil

Kasus	Jumlah Data	Dist. *Noise*	IPGNN	BP	RBF
1	40	N(0,0.1)	0.5265	0.3622	0.3673
2	40	N(0,0.1)	0.5718	0.2785	0.4741
3	40	N(0,0.1)	0.6082	0.3645	0.4056
4	40	N(0,0.2)	0.6706	0.2388	0.4507
5	40	N(0,0.2)	0.6250	0.3099	0.4215
6	40	N(0,0.2)	0.5761	0.2416	0.3307
7	40	N(0,0.3)	0.7321	0.2314	1.0656
8	40	N(0,0.3)	0.5102	0.3278	0.5577
9	40	N(0,0.3)	0.6315	0.2604	1.4034

5. PENUTUP

Pada tulisan ini kami melakukan suatu evaluasi eksperimental terhadap *incremental projection generalizing neural networks* (IPGNN) dalam memecahkan masalah aproksimasi pada ruang polinomial trigonometri. Berdasarkan hasil simulasi yang kami peroleh, kami mendapatkan kondisi sebagai berikut:

- Model yang optimal, yaitu order basis dari ruang *reproducing kernel Hilbert* yang digunakan, sangat bergantung pada jumlah data dan variansi noise dari data pembelajaran. Akibatnya, untuk mendukung supaya IPGNN dapat bekerja optimal maka dibutuhkan juga suatu metode pemilihan basis yang *incremental* juga. Sampai saat ini, metode pemilihan basis yang berkaitan dengan IPGNN yang bersifat *incremental* masih belum dikembangkan. Pemilihan model dengan menggunakan *subspace information criterion* (SIC) yang dikembang oleh Sugiyama & Ogawa masih untuk pembelajaran yang bersifat batch.
- Jumlah *neuron* pada lapisan tersembunyi secara teori akan lebih kecil atau sama dengan dimensi dari ruang *Reproducing Kernel Hilbert* yang digunakan. Pada simulasi yang kami peroleh ternyata kondisi ini sering tidak dipenuhi. Jumlah *neuron* tersebut cendrung bergantung pada jumlah data pembelajaran.
- Pada studi komparasi terhadap kapabilitas generalisasi yang dihasilkan oleh IPGNN dengan jaringan lain yang umum digunakan, yaitu *on-line backprogation* (BP) dan *growing radial basis function* (RBF), kami mendapatkan kondisi bahwa IPGNN tidak selalu memberikan kapabilitas generalisasi yang terbaik. IPGNN masih memberikan kapabilitas generalisasi terbaik ketika jumlah data pembelajaran cukup kecil atau variansi noise cukup besar. Selain itu, IPGNN tidak selalu memberikan hasil yang terbaik. Bahkan untuk kondisi dimana jumlah data pembelajaran cukup besar dan variansi noise cukup kecil, IPGNN selalu memberikan hasil yang lebih buruk dari BP.

REFERENSI

[1] Chen, S., C.F.N. Cowan and P. M. Grant. *Orthogonal Least Squares Learning Algorithm for Radial Basis Functions Networks*. IEEE Transaction on Neural Networks, vol. 2, no. 2, pp. 302-309, 1991

[2] Demuth, H. and M. Beale. *Neural Network Toolbox 4.0 User's Guide for Matlab*. The MathWorks Inc., 2000

[3] Kadirkamanathan. V. and M. Nirajan. *A Function Estimation Approach to Sequential Learning with Neural Networks*. Neural Computation 5, 954-975, 1993

[4] Ogawa, H. *Neural Networks Learning, Generalization and Over Learning*. Proceedings of the ICIIPS'92, International Conference on Intelligent Information Processing System, Beijing, China, 1992

[5] Platt, J. *A Resource Allocating Networks for Function Interpolation*. Neural Computation, vol. 3, no. 2, 213-225, 1991

[6] D.E. Rumelhart, G.E. Hinton and R.J. Williams, *Learning Internal Representation by Error Propagation*. Paralel Distributed Processing: Explorations in the Microstructure of Cognition, pp. 318-362, The MIT Press, Cambridge, MA, 1986

[7] Sugiyama, M. and H. Ogawa. *Exact Incremental Projection Learning in Neural Networks*. IEICE Technical Report, NC98-97, pp. 149-156, 1999

[8] Sugiyama, M. and H. Ogawa. *Incremental Projection Learning for Optimal Generalization*. Neural Networks, vol. 14, no. 1, pp. 53-66, 2001

[9] Sugiyama, M. and H. Ogawa. *Properties of Incremental projection learning*. Neural Networks, vol. 14, no. 1, pp. 53-66, 2001

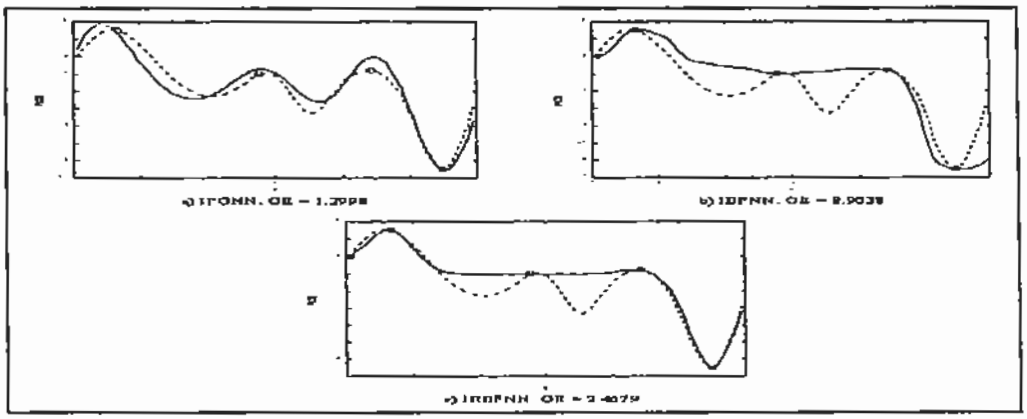
[10] Sugiyama, M. and H. Ogawa. *Subspace Information Criterion for Model Selection*. Neural Computation, vol.13, no.8, pp.1863-1889, 2001

[11] Sugiyama, M and H. Ogawa. *Incremental Projection Learning for Optimal Generalization in Neural Networks*. (Submitted). Also as Technical Report, Departmen of Computer Science, Tokyo Institute of Technology, Japan

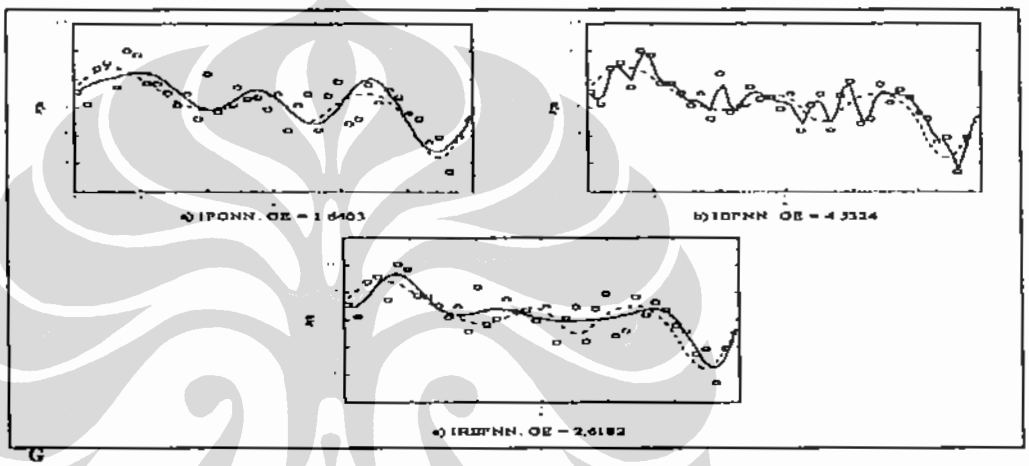
[12] Sugiyama, M and H. Ogawa. *Active learning for optimal generalization in trigonometric polynomial model*. To appear in IEICE Transactions on Fundamental of Electronic, Communication and Computer Science

[13] Yingwei, Y., N. Sundararajan, P. Saratchandran. *A Sequential Learning Schema for Function Interpolation Using Minimal Radial Basis Function Networks*. Neural Computation, vol. 9, 461-478, 1997

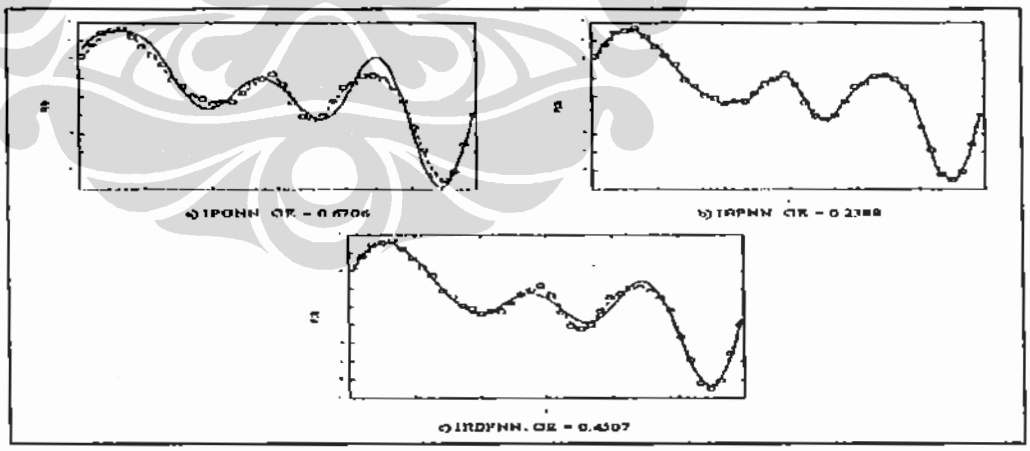
Lampiran:



Gambar L.1. Simulasi grafis untuk Kasus 10 dengan lima data pembelajaran dari Tabel 4. Garis solid dan putus-putus adalah fungsi target dan fungsi hasil pembelajaran. o adalah data pembelajaran.



Gambar L.2. Simulasi grafis untuk Kasus 5 dari Tabel 5. Garis solid dan putus-putus adalah fungsi target dan fungsi hasil pembelajaran. o adalah data pembelajaran.



Gambar L.3. Simulasi grafis untuk Kasus 4 dari Tabel 6. Garis solid dan putus-putus adalah fungsi target dan fungsi hasil pembelajaran. o adalah data pembelajaran